State University of Londrina
Center of Technology and Urbanism
Department of Electrical Engineering

Caio Henrique Azolini Tavares

Machine Learning Applied to Cooperative Spectrum Sensing in Cognitive Radios

Londrina, PR
2019

State University of Londrina

Center of Technology and Urbanism
Department of Electrical Engineering

Caio Henrique Azolini Tavares

Machine Learning Applied to Cooperative Spectrum Sensing in Cognitive Radios

Dissertation submitted to the Graduate Program in Electrical Engineering of the State University of Londrina in fulfillment of the requirements for the Degree of Master in Electrical Engineering.

Focus area: Electronic Systems
Specificity: Telecommunications Systems

Supervisor:
Prof. Dr. Taufik Abrão

Londrina, PR
2019

Ficha Catalográfica

Azolini Tavares, Caio Henrique

Machine Learning Applied to Cooperative Spectrum Sensing in Cognitive Radios. Londrina, PR, 2019. 40 p.

Dissertation (Master's Degree) – State University of Londrina, PR. Department of Electrical Engineering .

1. Telecommunication Systems. 2. Machine Learning. 3. Spectrum Sensing I. State University of Londrina. Department of Electrical Engineering . II. Machine Learning Applied to Cooperative Spectrum Sensing in Cognitive Radios.

Caio Henrique Azolini Tavares

Machine Learning Applied to Cooperative Spectrum Sensing in Cognitive Radios

Dissertation submitted to the Graduate Program in Electrical Engineering of the State University of Londrina in fulfillment of the requirements for the Degree of Master in Electrical Engineering.

Focus area: Electronic Systems
Specificity: Telecommunications Systems

Examining Board

Prof. Dr. Taufik Abrão
Department of Electrical Engineering
State University of Londrina
Supervisor

Prof. Dr. Mario Lemes Proença Jr.
Department of Computing
State University of Londrina

Prof. Dr. José Carlos Marinello Filho
Department of Electrical Engineering
State University of Londrina

May 13th, 2019

"A wizard is never late, nor is he early. He arrives precisely when he means to"

Gandalf

Acknowledgments

I would like to thank my supervisor Taufik Abrão for his incredible patience and teachings. My parents and my wife Rebecca for the support; Lastly, to the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* for the financial support (Finance Code 001).

Resumo

Nesta dissertação discute-se a tarefa de sensoriamento espectral cooperativo em rádios cognitivos baseada em ferramentas estatísticas e de aprendizado de máquina. Considerando o uso de detectores de energia nos usuários secundários (SU), compara-se a aplicação de diferentes ferramentas de tomada de decisão como forma de combinar o nível de energia obtido em cada SU. Através de análise estatística do nível de energia, o modelo proposto *Weighted Bayesian* combina as probabilidades *a posteriori* de ocupação de canal obtidas em cada SU, ponderadas pela respectiva taxa de sinal-ruído (SNR), de forma a obter uma decisão que, por sua vez, é transmitida de volta aos usuários secundários. Com o objetivo de suprimir a necessidade de obter e transmitir o nível de SNR nos SUs, compara-se a aplicação de três modelos de aprendizado de máquina, a saber *naive Bayes*, rede neural *feed-forward* e máquina de vetor de suporte (SVM), como forma de estimar a probabilidade *a posteriori* de ocupação do canal baseada somente no nível de energia e um conjunto de treinamento. Resultados numéricos mostraram que todos os três modelos obtiveram uma performance próxima da ótima definida pela técnica *maximum ratio combining*, especialmente a técnica SVM com *kernel* linear. Por fim, o objetivo desta dissertação é prover uma análise abrangente de diferentes técnicas de tomada de decisão em cenários realistas de sensoriamento espectral cooperativo, considerando a complexidade computacional de se treinar modelos de aprendizado de máquina.

Palavras-chave: Radios cognitivos; sensoriamento espectral; aprendizado de máquina;

Abstract

In this dissertation we tackled the task of cooperative spectrum sensing on cognitive radio networks based on statistical and machine learning tools. Considering the use of energy detectors on the secondary users (SU), we compared the application of different decision-making tools as a way to combine the energy level obtained at each SU. Through statistical analysis of the estimated energy levels, the proposed Weighted Bayesian model combines the *a posteriori* probabilities of channel occupancy obtained at each SU weighted by their respective signal-to-noise (SNR) ratio in order to obtain a decision, which gets transmitted back to the SUs. In order to suppress the need of obtaining and transmitting the SNR level on the SUs, we compared the application of three machine learning models, namely naive Bayes, feed-forward neural network and support vector machine (SVM) as a way of estimating the *a posteriori* probability of channel occupancy based solely on the energy level and a training dataset. Numerical results have demonstrated that all three models perform close to the optimum maximum ratio combining (MRC) technique, specially the SVM technique with linear kernel. Ultimately, the goal of this dissertation is to arrive at an in-depth analysis of different decision-making tools in a realistic cooperative spectrum sensing scenario, considering the computational complexity of training machine learning models.

Keywords: Cognitive radios; spectrum sensing; machine learning;

Contents

List of Figures

List of Tables

List of Acronyms

CR Cognitive radio

RF Radio frequency

SU Secondary user

PU Primary user

ML Machine learning

SVM Support Vector Machine

GMM Gaussian Mixture Model

NBC Naive Bayes Classifier

MLP Multilayer perceptron

MRC Maximum Ratio Combining

AuC Area under the curve

ROC Receiver operating characteristic

SNR Signal-to-noise ratio

AWGN Additive white Gaussian noise

SS Spectrum sensing

WB Weighted Bayesian

GM Gaussian mixture

CRN Cognitive radio network

NLOS Non-line-of-sight

KKT Karush-Kuhn-Tucker

QP Quadratic programing

LLF Log likelihood function

MCS Monte-Carlo simulation

Conventions and List of Symbols

The following conventions were used in the notation of equations and formulas:

- bold lowercase letters are vectors, example: $\mathbf{x}, \mathbf{d}$;
- bold uppercase letters are matrices, example: $\mathbf{W}$;
- x_i is the i_{th} element of vector $\mathbf{x}$;
- y_{ij} is the element at i_{th} row and j_{th} column of matrix $\mathbf{Y}$;
- $\mathcal{N}(m, \sigma^2)$ is a random process with normal distribution of mean m and variance σ^2;
- $\Gamma(x, y)$ is a random process with Gamma distribution of shape x and scale y;
- $Q(\cdot)$ is the right tail probability function;
- $||\cdot||$ is the Euclidean norm of a vector;
- $\{\cdot\}^T$ is the matrix transpose operator;

1 Introduction

Cognitive radios (CR) systems are a proposed solution to the spectrum scarcity problem found in radio frequency (RF) environment that aims to improve the overall spectrum utilization. Several studies showed (CHEN; OH, 2014) that licensed spectrum bands are often not occupied by the licensed users, thus creating the opportunity for other devices to access the unoccupied spectrum in an opportunistic way. These opportunistic devices, denoted as secondary users (SU) in the context of cognitive radios, need to be able to sense the spectrum to assess the presence or absence of licensed users, denoted as primary users (PU), either individually or cooperatively.

The idea of cognitive radios was first introduced by Joseph Mitola III in 1999 (MITOLA; MAGUIRE, 1999), but has been given a lot of attention recently due to the proposed heterogeneous nature of 5G networks (TSENG et al., 2015; JIA et al., 2016; CHAE; JEONG; LEE, 2018; LI et al., 2018a).

Spectrum sensing in cognitive radios still poses a challenge for high-performance and low-energy systems due to the fact that performance is often proportional to the spectrum sensing period which, in turn, is an energy consuming task that also degrades the spectral efficiency of the secondary users (since they need to spend time and energy on a task that does not effectively result in transmitted bits).

Generally speaking, cooperative spectrum sensing schemes falls into two topologies: distributed (LI; YU; HUANG, 2009) or centralized (MA; ZHAO; LI, 2008). Centralized approaches require the SUs to transmit information regarding the local spectrum sensing (e.g., the SU binary decision on spectrum occupancy for hard combination schemes) to a fusion center, which in turn combines the received data according to a given method, decides on the spectrum occupancy and retransmits the decision back to the SUs. On the other hand, distributed approaches relies on information sharing among neighboring SUs and consensus methodologies, thus eliminating the need for a fusion center.

Since the task of determining the channel status based on spectrum sensing is clearly a classification task, several authors considered the use of machine learning models as inference tools. Machine learning, in turn, is a way of programming computers to optimize a performance criterion using example data or past experience (ALPAYDIN, 2014). This apparent self-learning characteristic is by itself mostly based on applied statistics, whereas the training and inference capabilities owe their efficiency to great computer science algorithms.

In (THILINA et al., 2013), the authors propose and compare the performance of several supervised and unsupervised machine learning (ML) techniques for cooperative spectrum sensing, such as Support Vector Machines (SVM), the K-means clustering algorithm and Gaussian Mixture Model (GMM), but do not provide a comparison of detection performance over different training set sizes. In (AZMAT; CHEN; STOCKS, 2015), the authors study the use of machine learning algorithms for spectrum occupancy in cognitive radio networks, which include the Naive Bayesian Classifier (NBC). In (LI et al., 2018b), the authors propose user grouping algorithms to improve spectrum sensing results and SVM training time and in (BKASSINY; LI; JAYAWEERA, 2012) the authors enumerate the pros and cons of several unsupervised and supervised machine learning techniques applied to spectrum sensing, such as the requirement of data labeling for supervised models and the risk of overfitting.

In this dissertation, we explored the use of several statistical and machine learning tools within two papers addressing centralized cooperative spectrum sensing. In the first, entitled *Bayesian Estimators for Cooperative Spectrum Sensing in Cognitive Radio Networks*, a conference paper published in the IEEE URUCON' 2017 conference, we made use of the statistical modeling of the estimated energy level on SUs in order to derive two analytical Bayesian methodologies that aimed at reducing the overhead imposed by the maximum ratio combining (MRC) technique.

The second paper developed in the context of this Dissertation, entitled *Machine Learning-based Models for Spectrum Sensing in Cooperative Radio Networks*, which is an extension of the first work. Currently, the manuscript is under review in the IET Communications journal. In this work, the application of three popular supervised machine learning models (multilayer perceptron, support vector machine and naive Bayes) was considered to the task of channel status inference based on cooperative energy detection spectrum sensing.

The performance of the models are compared to the optimum MRC technique through area under the curve (AuC) metrics and receiver operating characteristic (ROC) graphics, under additive white Gaussian noise (AWGN) and Rayleigh channel models. In addition, the computational performance of each model is evaluated through standard profiling tools in order to draw a complexity trade-off analysis. Furthermore, an analysis of training set size is conducted to reveal the effects on channel detection and training time.

2 Development

Throughout the development of both papers, we consider centralized spectrum sensing based on energy detection. With the energy information gathered on the secondary users, we employ and compare different combination techniques based on analytical and machine learning models.

The basic framework upon which every technique is designed is the decision between two hypothesis H_1 (channel is occupied by the primary user) and H_0 (channel is free), which can be written as:

$$\begin{cases} H_1 : z_i(k) = h_i x(k) + n_i(k), \text{ if the PU is active} \\ H_0 : z_i(k) = n_i(k), \text{ otherwise} \end{cases} \tag{2.1}$$

where $z_i(k)$ is the signal at the secondary user, $n_i(k)$ is the zero-mean Gaussian channel noise with variance σ_n^2, $x(k)$ is the signal transmitted by the primary user and $h_i(k)$ is the channel gain measured from the primary user to the i_{th} secondary user, which can be specified by the path loss and fading components.

From this, the estimated energy can be defined as:

$$y_i = \frac{1}{\sigma_n^2} \sum_{k=1}^{K} z_i(k)^2 \tag{2.2}$$

Based on the definition above, we can write the basic modeling for each of the techniques.

2.1 Analytical Techniques

The analytical techniques explored throughout A.1 and A.2 are based on knowledge about the communication channel in use during the spectrum sensing periods. This knowledge include the signal-to-noise ratio at the secondary users and the *a priori* probability of channel occupancy.

2.1.1 Maximum Ratio Combining

The maximum ratio combining technique, considered as the upper-bound in channel status detection performance and explored in A.1 and A.2, is an analytical technique in which the energy level sensed on the secondary users are combined proportionally to their current signal-to-noise levels. In this sense, the estimated channel status $\hat{S}$ can be written as:

$$\hat{S} = \begin{cases} H_1, & \text{if } \sum_{i=1}^{N} w_i y_i \geq \lambda \\ H_0, & \text{otherwise} \end{cases} \tag{2.3}$$

where $w_i = \frac{\gamma_i}{\sum_N \gamma_i}$, γ_i is the SNR level on the i_{th} secondary user and λ is a constant used to reflect the desired false-alarm probability of the estimator.

2.1.2 Weighted Bayesian

The weighted Bayesian technique is an analytical technique developed on A.1, which makes use of the statistical analysis of the estimated energy level on the secondary users in order to reduce the data overhead imposed by the MRC approach. The goal is to map the estimated energy level to a *posteriori* probability of channel occupancy in the secondary user and then send it to the fusion center for combination.

The occupancy probability can be defined as:

$$P(H_1|y_i) = \frac{f_{\Gamma_{\frac{K}{2}}}(y_i)P(H_1)}{f_{\mathcal{X}^2_N}(y_i)P(H_0) + f_{\Gamma_{\frac{K}{2}}}(y_i)P(H_1)} \tag{2.4}$$

where $f_{\Gamma_{\frac{K}{2}}}$ and $f_{\mathcal{X}^2_N}$ are the probability density functions defined for hypothesis H_1 and H_0, respectively. They are discussed in details at A.1.

From Eq. (2.4), the estimated channel status can be defined as:

$$\hat{S} = \begin{cases} H_1, & \text{if } P(H_1|\mathbf{y}) \geq 1 - P^*_{\text{fa}} \\ H_0, & \text{otherwise} \end{cases} \tag{2.5}$$

2.1.3 Gaussian Mixture

The Gaussian mixture model, also explored in A.1, is an analytical technique based on the assumption of Gaussianity and independence among the energy samples obtained on each individual secondary user on the network. These two assumptions allows us to completely specify the mean $\boldsymbol{M}$, variance $\boldsymbol{\Sigma}^2$ and mixture proportions $\boldsymbol{\pi}$ as:

$$
\begin{aligned}
\boldsymbol{M} &= \begin{bmatrix} K & \dots & K \\ K(1+\bar{\gamma}_i) & \dots & K(1+\bar{\gamma}_N) \end{bmatrix} \\
\boldsymbol{\Sigma}^2 &= \begin{bmatrix} 2K & \dots & 2K \\ 2K(1+\bar{\gamma}_i)^2 & \dots & 2K(1+\bar{\gamma}_N)^2 \end{bmatrix} \\
\boldsymbol{\pi} &= \begin{bmatrix} P(H_0) & P(H_1) \end{bmatrix}^T
\end{aligned}
\tag{2.6}
$$

where K is the number of samples obtained during a sensing period.

Once the statistical aspects of the energy samples are defined, the estimated channel status can be written as:

$$
\hat{S} = \begin{cases} H_1, & \text{if } P(H_1|\mathbf{y}) \geq 1 - P_{\text{fa}}^{*} \\ H_0, & \text{otherwise} \end{cases} \tag{2.7}
$$

2.2 Machine Learning Techniques

The methods discussed in Section 2.1 are based on previous knowledge about the cognitive radio channel, such as instantaneous and average signal-to-noise ratio at the SUs and the *a priori* probability of channel occupancy $P(H_1)$.

The following section explores the use of techniques which do not require any previous knowledge of the channel, with the necessary counterpart of cooperation from the primary user, which should provide the fusion center with labeled samples matching a number of previous sensing periods.

Figure 2.1 depicts a typical cognitive radio scenario with the cooperation from the primary user. The vector $\mathbf{d}$ consists of M channel status results (referring to either hypothesis H_0 or H_1) and is used by the fusion center in order to train machine learning models.

The training process is an automated way to approximate a function which maps the estimated energy samples on the SUs to a label provided by the PU.

After the training phase, the models are able to decide the channel status based on unseen energy samples. This allows the design of the cognitive radio network to consider the best-effort approach, either by defining the cooperation of the primary user or performing a way to provide channel knowledge into the secondary users.

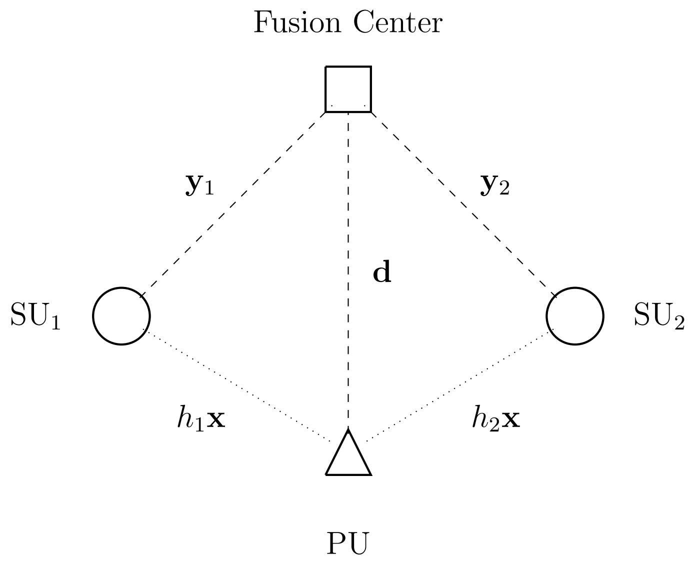


Figure 2.1: Illustration of a cognitive radio network and the interactions between SUs and PU. The vector **d** contains previous channel status and is used by the fusion center in order to perform training of supervised machine learning models.

2.2.1 Naive Bayes

The naive Bayes classifier, explored in A.1 and A.2, is based on the same assumptions defined at 2.1.3 and can estimate the channel occupancy *a posteriori* probability given the detected energy $\mathbf{y}$ through Bayes Theorem:

$$P(H_1|\mathbf{y}) = \frac{f(\mathbf{y}|H_1)P(H_1)}{f(\mathbf{y}|H_1)P(H_1) + f(\mathbf{y}|H_0)P(H_0)} \tag{2.8}$$

where $P(H_1)$ and $P(H_0)$ are the *a priori* probabilities of each hypothesis, estimated as follows:

$$P(H_1) = \frac{M^{\{H_1\}}}{M} \tag{2.9}$$

$$P(H_0) = \frac{M^{\{H_0\}}}{M} \tag{2.10}$$

where $M^{\{H_i\}},\ i = 0, 1$ is the number of the ith hypothesis occurrences.

Once the posteriori probability has been defined, the estimated channel status is given by:

$$\hat{S}_{\text{NB}} = \begin{cases} H_1, & \text{if } P(H_1|\mathbf{y}) \geq 1 - P^*_{\text{fa}} \\ H_0, & \text{otherwise} \end{cases} \tag{2.11}$$

2.2.2 Multilayer Perceptron

The multilayer perceptron is a well-known and widely adopted machine learning technique explored in A.2. Its basic functionality can be described as a non-linear function approximator. The set of inputs provided at the first layer of the network flow through a series of inner (or hidden) layers, where each node applies a non-linear sigmoidal function to the weighted combination of inputs.

Figure 2.2 depicts a typical network with one hidden layer of J neurons, I inputs and Θ outputs.

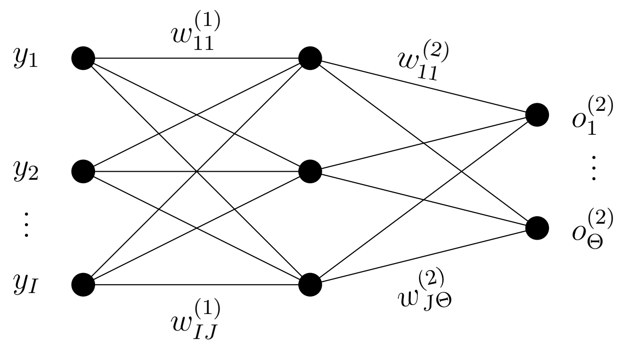


Figure 2.2: Illustration of a feed-forward neural network with one hidden layer

During the training procedure, the set of weights w are adjusted according to the deviation of the actual outputs to expected outputs.

For the spectrum sensing problem, we consider a neural network with one hidden layer, one output unit, i.e. $\Theta = 1$ (since we are interested in binary classification) and with the number of inputs equal to the number of SUs, i.e. $I = N$. Also, we consider the number of neurons of the hidden layer to be equal to the number of inputs N, so $J = N$. Therefore, we can write the output of any neuron in the hidden layer as:

$$o_j^{(1)} = \sigma\left(\sum_{i=0}^{N} w_{ij}^{(1)} y_i\right) \tag{2.12}$$

and similarly for the output neurons:

$$o_\theta^{(2)} = \sigma\left(\sum_{j=0}^{N} w_{j\theta}^{(2)} o_j^{(1)}\right) \tag{2.13}$$

in both cases, y_0 and $o_0^{(1)}$ are known as bias inputs and are equal to 1. They are necessary in order to shift the activation function away from the origin.

The output of the network is mapped to the estimated channel status according to the following:

$$\hat{S}_{\text{MLP}} = \begin{cases} H_1, & \text{if } o \geq 1 - P_{\text{fa}}^* \\ H_0, & \text{otherwise} \end{cases} \tag{2.14}$$

where P_{fa}^* is the target false alarm probability.

2.2.3 Support Vector Machine

The support vector machine is a maximum margin classifier explored in A.2. Putting simply, the SVM approximates a hypothesis function $h(\mathbf{y})$ whose output is positive for hypothesis H_1 and negative for H_0.

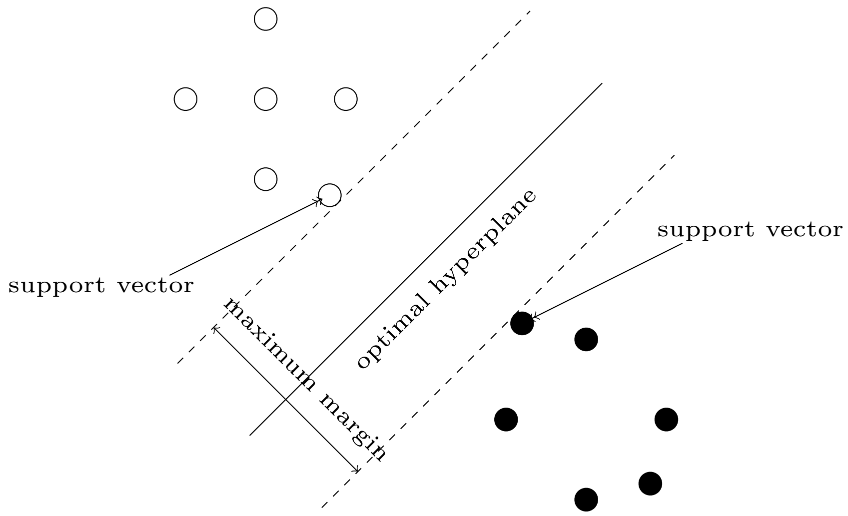


Figure 2.3: Illustration of a support vector machine decision plane. The optimal hyperplane separates the two classes (filled circles and empty circles) at half distance of the maximum margin defined by the support vectors.

After solving an optimization problem to find the support vectors among the training samples (i.e. the samples which define the margin between the two hypothesis in feature space), the output of the SVM can be written as:

$$h(\mathbf{y}) = \sum_{m=1}^{M} \alpha_m^\star d_m \kappa(\mathbf{y}, \mathbf{y}_m) + b^\star \tag{2.15}$$

Once the output is obtained, a mapping function is used to convert the scalar to a *posterior* probability of occupancy. Finally, the estimated channel status is obtained by:

$$\hat{S}_{\text{SVM}} = \begin{cases} H_1, & \text{if } \hat{P}(h(\mathbf{y})) \geq 1 - P_{\text{fa}}^{*} \\ H_0, & \text{otherwise} \end{cases} \tag{2.16}$$

3 Numerical Results

In the following sections, the numerical results obtained at each of the annexes will be discussed separately, considering the techniques discussed in Chapter 2.

3.1 Bayesian Estimators for Cooperative Spectrum Sensing

On A.1, two cognitive radio scenarios were used to compare the techniques discussed in Sections 2.1.1, 2.1.2, 2.1.3 and 2.2.1. In both cases, the radio channel was assumed to be AWGN.

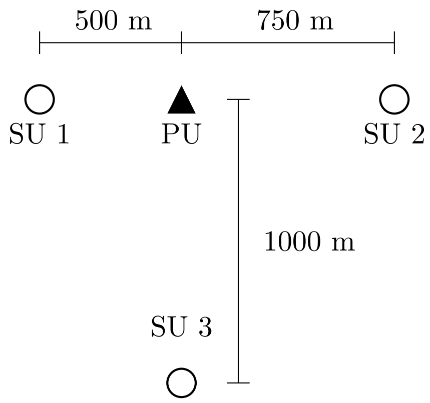


Figure 3.1: First cognitive radio scenario used in A.1

The first scenario depicted at Fig. 3.1 has three secondary users and one primary user, separated by distances of 500m, 750m and 1000m. The spectrum sensing results obtained by each technique were compared according to receiver operating characteristic (ROC) curves and area under the curve (AuC) metrics. The complete channel and simulation configuration can be found on A.1.

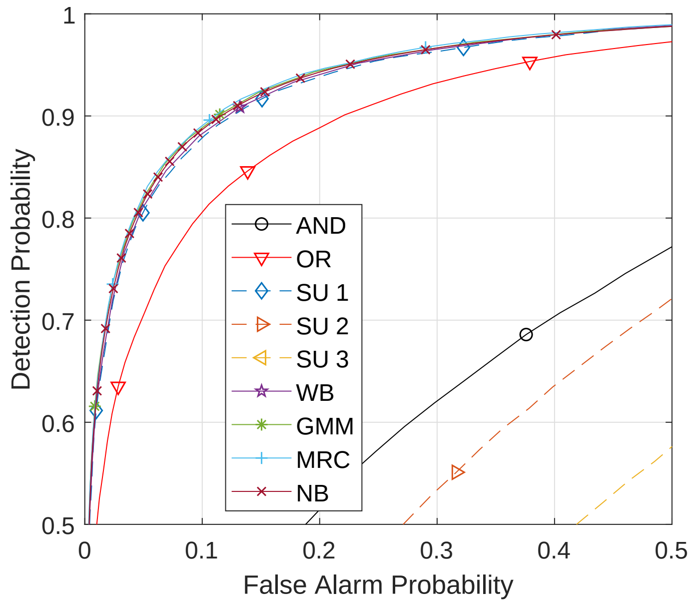


Figure 3.2: ROC curve for scenario I

The ROC curves on Fig. 3.2 show very similar graphical results obtained by the considered techniques. Clearly, the analytical and machine learning techniques greatly outperform the simple methods such as OR and AND and the individual results of the secondary users (not considering cooperation).

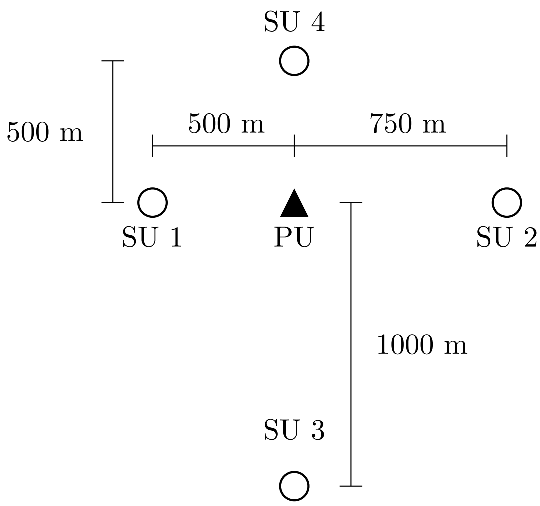


Figure 3.3: Second cognitive radio scenario used in A.1

The second scenario considered on A.1 is depicted at Fig. 3.3, the main difference from scenario I is the addition of a fourth secondary user also at 500m of the primary user. This addition provides additional energy samples with relatively high SNR to the fusion center, which is expected to improve the detection performance.

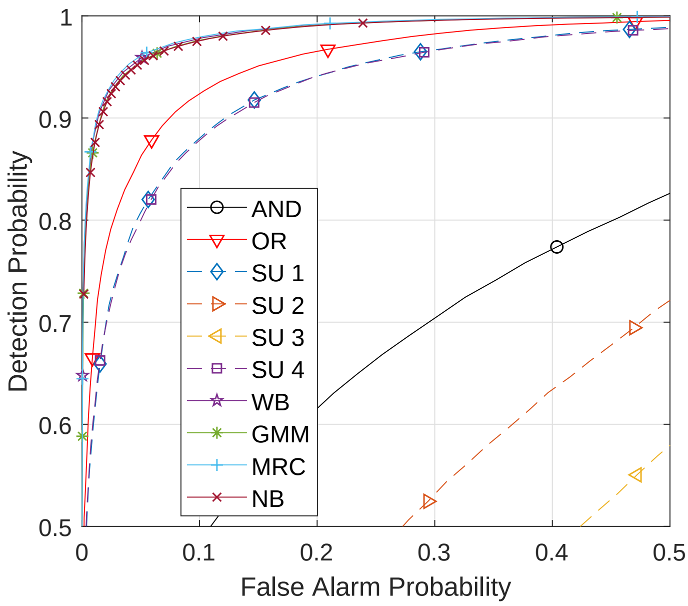


Figure 3.4: ROC curve for scenario II

Indeed, from Fig. 3.4 the performance of all techniques is clearly increased, surpassing the 90% detection probability mark at 10% of false alarm.

Since all techniques have shown similar results graphically, Table 3.1 summarizes the obtained AuC metrics for both scenarios, showing the MRC as the upper-bound, followed by GM in scenario I and WB in scenario II.

Table 3.1: Monte Carlo simulation results (AuC).

Scenario	MRC	GM	WB	NB	AND	OR
Scenario I	0.9627	0.9615	0.9603	0.9608	0.7104	0.9314
Scenario II	0.9913	0.9902	0.9909	0.9891	0.7735	0.9716

3.2 Machine Learning Models for Cooperative Spectrum Sensing

On A.2, the scenario from Fig. 3.1 was used to compare the techniques 2.1.1, 2.2.1, 2.2.2 and 2.2.3 under AWGN and Rayleigh fading channels. We compared the results of channel status inference using ROC curves and AuC metrics. Additionally, computational performance measurements were extracted during the training and inference phases of each model, to better reflect the actual cost of implementation.

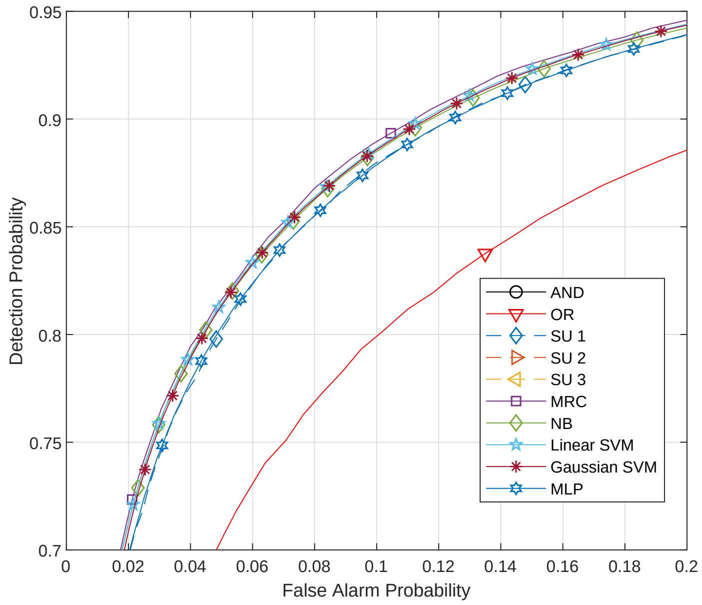


Figure 3.5: ROC curve for AWGN channel

On Fig. 3.5, we obtained the ROC curve for the aforementioned techniques operating in an AWGN channel. MRC is established as an upper-bound followed closely by the SVM with linear kernel.

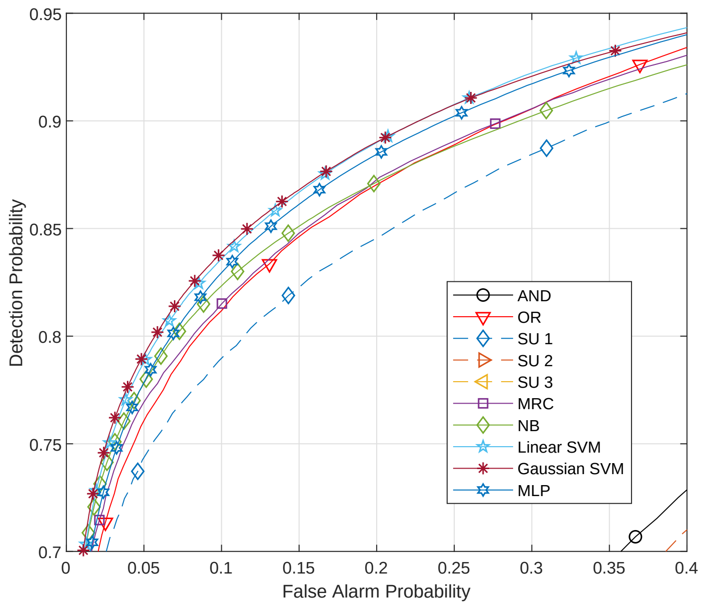


Figure 3.6: ROC curve for Rayleigh fading channel

On the other hand, when operating in Rayleigh fading channel, the MRC is no longer the upper-bound due to the lack of channel estimation. In this case, both the linear and Gaussian kernel SVMs establish the best channel detection performance.

3.2.1 Computational Costs

One interesting metric to take into account during implementation of machine learning models is the size of the training set, which can directly impact the training duration as well as the final model accuracy.

Table 3.2 summarizes the training and inference time averaged over 20 rounds, for a training set of 500 samples. Whereas on Fig. 3.7, the time spent during training phase is shown for 5 different training set sizes, from 50 to 1000 samples.

Table 3.2: Average time spent during training/inference phase for 500 training samples

SS Technique	**Training** [sec]	**Inference** [sec]
NB	0.420	0.141
SVM-Linear	1.060	0.035
SVM-Gaussian	0.327	0.082
MLP	**0.230**	0.024
MRC	-	**0.006**

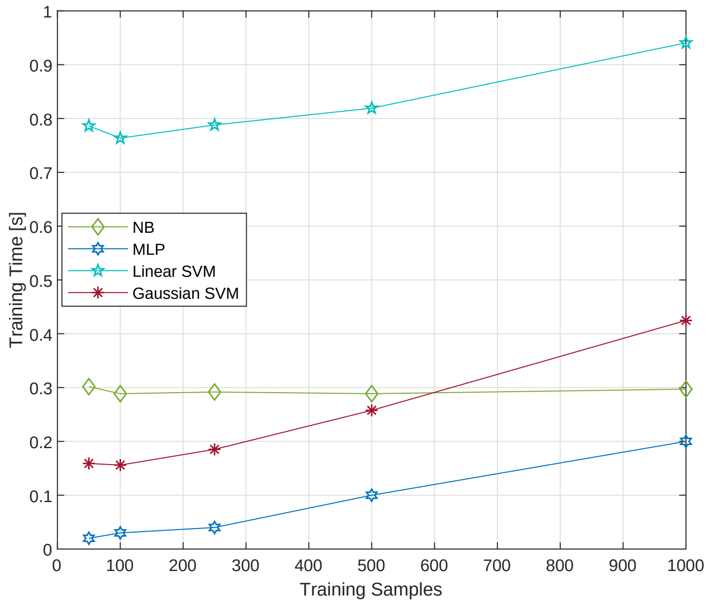


Figure 3.7: Time spent during training phase for different training set sizes.

Finally, on Fig. 3.8 we have the difference in AuC resulting from the variation of the training set size. Clearly, the sharpest difference in performance is evident from 50 to 500 samples.

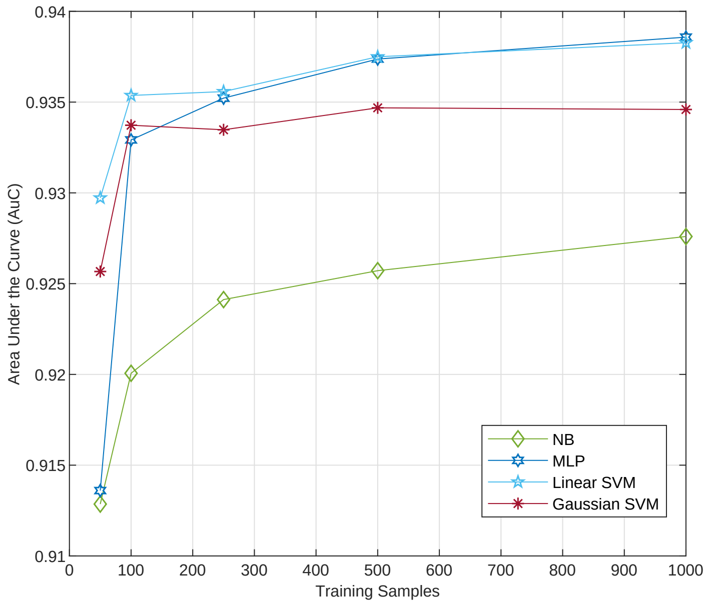


Figure 3.8: AuC for different training set sizes.

One interesting aspect that has not been taken into account by the metrics and graphics above is the implementation considerations of each technique.

Despite having no training required and practically instant inference time, the MRC technique requires the propagation of SNR information from SUs to the fusion center, as well as channel estimation in realistic channels with fading, which can pose a difficulty to implement.

On the other hand, each and every supervised machine learning technique such as the ones described by 2.2 requires the cooperation from the PU to provide the fusion center with channel status samples for the training of the models.

3.3 Simulation Code

The code used for the simulations performed in A.1 and A.2 is open source and can be found at `https://github.com/caiotavares/spectrum-sensing`. Below are listed some of the algorithms used during the development of A.2.

Listing 3.1: Naive Bayes implementation in MATLAB

```
function NB = NB(train, test)

NB.model = fitcnb(train.X,train.Y);
[~,NB.P,~] = predict(NB.model,test.X);
NB.positiveClass = 2;
NB.name = 'NB';
```

Listing 3.2: Linear SVM implementation in MATLAB

```
function SVM = LSVM(train, test)

scoreModel = fitcsvm(train.X,train.Y,'KernelFunction', 'linear');
SVM.model = fitPosterior(scoreModel); % Transforms score to
    posterior probability
[~,SVM.P] = predict(SVM.model,test.X);
SVM.positiveClass = 2;
SVM.name = 'LSVM';
```

Listing 3.3: Gaussian SVM implementation in MATLAB

```
function SVM = GSVM(train, test)

scoreModel = fitcsvm(train.X,train.Y,'KernelFunction', 'gaussian'
    );
SVM.model = fitPosterior(scoreModel); % Transforms score to
    posterior probability
[~,SVM.P] = predict(SVM.model,test.X);
SVM.positiveClass = 2;
SVM.name = 'GSVM';
```

Listing 3.4: Multilayer perceptron implementation in R using the RSNNS library

```
MLP <- function ()
{
  library(R.matlab)
  library(RSNNS)
  data <- R.matlab::readMat("ss.mat")$ML

  X.train <- data.frame(data[,,1]$train[1])
  X.train <- cbind(X.train, data[,,1]$train[2])
  colnames(X.train)[ncol(X.train)] <- "Status"
  X.train$Status <- factor(X.train$Status, levels = c(0,1))

  inputs = ncol(X.train)-1
  hiddenUnits = inputs

  model <- RSNNS::mlp(x = X.train[, -ncol(X.train)], y = RSNNS::
      decodeClassLabels(X.train$Status), size = hiddenUnits,
      hiddenActFunc = "Act_Logistic")
  weights = weightMatrix(model)
  W_hidden <- weights[1:inputs,(inputs+1):(inputs+hiddenUnits)]
  W_output <- weights[(inputs+1):(inputs+hiddenUnits),(inputs+
      hiddenUnits+1):(inputs+hiddenUnits+2)]
```

```
  bias <- extractNetInfo(model)$unitDefinitions$unitBias
  R.matlab::writeMat(W_hidden = W_hidden, W_output = W_output,
     bias = bias, con = "MLP.mat")
}
```

4 Conclusions

Considering the work conducted on *Annex A*, it was shown that the proposed Bayesian methods perform well on simple AWGN channel models, and can be considered as alternatives to the optimum MRC technique. Also, by comparing different training set sizes, it was possible to show that the Naive Bayes model performance is upper-bounded by a Gaussian Mixture Model with diagonal covariance matrix.

On the other hand, on the work developed on *Annex B*, it was shown that all machine learning models perform very well when compared to the MRC technique under AWGN channel, with special attention on the Support Vector Machine with linear kernel function, which achieved the highest Area Under the Curve metric. Whereas under Rayleigh fading channel, all studied ML models achieved a high AuC than MRC, mostly due to the varying SNR on the SUs for each sensing period. In addition, by using standard profiling tools, the computational performance of each model was evaluated. Based on execution time metrics, the MLP was shown to have the best trade-off between computational complexity and AuC performance.

Both works provide insights into how well some common supervised machine learning models perform on the problem of spectrum sensing for cognitive radios under AWGN and Rayleigh channels and for different training set sizes.

Event though several authors explored the use of machine learning models in the context of cognitive radio spectrum sensing, we've shown that by changing the number of samples used for training, one can tune the trade-off between time spent in training phase and the performance obtained on channel status inference, which could affect each model differently.

Despite not being optimal, the time-based computational complexity obtained can be used as a heuristic for choosing which technique is more suitable for the desired implementation scenario.

In practice, the decision to which category and technique (analytical based on channel knowledge or machine learning based on PU cooperation) to apply on a cognitive radio network depends on the system requirements and capabilities. Despite being popular in many applications, supervised machine learning could require periodic training in order to avoid performance degradation (due to deviations on the channel characteristics over time), which could pose additional difficulties.

4.1 Future works

One interesting aspect that could be addressed in a future work is how often do the machine learning models need to be re-trained in a non-stationary channel, such as one imposed by SU mobility to avoid performance degradation. Also, varying numbers of PUs/SUs on the cognitive radio network could affect the channel status estimation.

Appendix A – Papers developed

A.1 Bayesian Estimators for Cooperative Spectrum Sensing

Title:	*Bayesian Estimators for Cooperative Spectrum Sensing in Cognitive Radio Networks*
Authors:	Caio Henrique Azolini Tavares and Taufik Abrão
Published:	December/2017
Journal/Conference:	*2017 IEEE URUCON*

Bayesian Estimators for Cooperative Spectrum Sensing in Cognitive Radio Networks

Caio Henrique Azolini Tavares and Taufik Abrão
Department of Electrical Engineering
State University of Londrina, Londrina, Parana, Brazil
Email: caio.tavares11@gmail.com; taufik@uel.br

***Abstract*—In this paper we consider centralized cooperative spectrum sensing (SS) techniques for cognitive radio networks using energy detector scheme. In light of the requirements imposed by centralized SS methods such as Maximum Ratio Combining (MRC), namely the estimation and transmission of the signal-to-noise ratio (SNR) on each secondary user, as well as the transmission of the exact energy level to the fusion center, we aim to analyze and compare alternative Bayesian techniques to tackle the trade-off between operation overhead and detection performance. Based on the statistics of classic energy detection scheme, we consider three Bayesian SS estimators, the Weighted Bayesian (WB) estimator, the Gaussian Mixture (GM) estimator and a Naive Bayes (NB) classifier. We compare the results of these techniques with well established cooperative SS schemes, such as the MRC, AND and OR rule. Through the use of Monte Carlo simulations, our results shows that the Bayesian techniques evaluated offer similar performance in terms of area under the ROC curve (AUC) regarding the optimum MRC technique, while requiring less operation overhead for the secondary users.**

***Keywords* – Cognitive Radios; Spectrum Sensing; Gaussian Mixture Models; Bayesian Inference; Machine Learning;**

I. INTRODUCTION

Cognitive radios (CR) systems are a proposed solution to the spectrum scarcity problem in radio frequency (RF) environment that aims to improve the overall spectrum utilization. Several studies showed [1] that licensed spectrum bands are often not occupied by the licensed users, thus creating the opportunity for other devices to access the unoccupied spectrum in an opportunistic way. These opportunistic devices, denoted as secondary users (SU), need to be able to sense the spectrum to assess the presence or absence of licensed users, denoted as primary users (PU), either individually or cooperatively. The idea of cognitive radios was first introduced by Joseph Mitola III in 1999 [2], but has been given a lot of attention recently due to the proposed heterogeneous nature of 5G networks [3].

Generally speaking, cooperative spectrum sensing schemes falls into two topologies: distributed [4] or centralized [5]. Centralized approaches require the SUs to transmit information regarding the local spectrum sensing (e.g., the SU binary decision on spectrum occupancy for hard combination schemes) to a fusion center, which in turn combines the received data according to a given method, decides on the spectrum occupancy and retransmit the decision back to the SUs. On the other hand, distributed approaches relies on information sharing among neighboring SUs, thus eliminating the need for a fusion center.

Within centralized cooperative spectrum sensing techniques, several authors explored the use of machine learning (ML) models for SU data combination and decision-making in the fusion center, such as [6] and [7]. Such models relies heavily on the statistical characterization of the detection scheme considered.

In this paper, we leverage the knowledge of the statistical model of energy detection scheme and analyze Bayesian estimators for centralized cooperative spectrum sensing, namely the Weighted Bayesian (WB), Learning-Based Naive Bayes (NB) and Gaussian Mixture model (GM). The motivation for studying such techniques is to reduce the overhead and complexity imposed by traditional well-established centralized cooperative techniques such as the Maximum Ratio Combining (MRC) [5].

With this in mind, the WB estimator translates the estimated energy on the SU to a channel occupancy probability before transmitting to the fusion center, which in turn is bounded, thus, requires less bytes to perform such task. On the other hand, the NB classifier suppresses the need to estimate and transmit the SNR on each SU by considering a machine learning approach, modeling the system probability distribution according to a training set of labeled energy levels.

The remainder of this paper is organized as follows. Section II discusses the system model for the cognitive radio network and the statistics of the primary user signal. Section III details the cooperative spectrum sensing techniques with soft combination schemes. Section IV discusses comparative results for such SS techniques through numeric results from Monte Carlo simulations. Finally, Section V offers the remarks and main conclusions.

II. SYSTEM MODEL

We consider a cognitive radio network (CRN) with one PU and n SUs, where each secondary user employs energy detection for spectrum sensing with sensing period τ and sensing bandwidth w. The sampling frequency is considered to be at the Nyquist rate, $f_s = 2w$, thus the SU acquires $K = 2w\tau$ samples at each sensing period.

We can write the signal at the i_{th} SU according to two hypothesis:

$$\begin{cases} H_1 : z_i(k) = h_i x(k) + n_i(k), \text{ if the PU is active} \\ H_0 : z_i(k) = n_i(k), \text{ otherwise} \end{cases} \quad (1)$$

where h_i is the channel coefficient from the PU to the i_{th} SU, $x(k)$ is the transmitted PU signal and $n_i(k)$ is the background thermal noise at the i_{th} SU receiver considered to be a zero-mean Gaussian random variable with variance σ_n^2. Furthermore, the channel coefficient h_i is described by the path-loss and fading components:

$$h_i = g_i \sqrt{d_i^{-\alpha}} \tag{2}$$

where g_i is the fading component considered to be unitary throughout this paper, d_i is the Euclidean distance between the PU and the i_{th} SU and α is the path-loss exponent, which we have considered as 4, indicating a non-line-of-sight (NLOS) channel environment.

At the end of the spectrum sensing period, the *estimated normalized energy* level at the i_{th} SU is given by:

$$y_i = \frac{1}{\sigma_n^2} \sum_{k=1}^{K} z_i^2(k) \tag{3}$$

where K is the number of samples deployed in the spectrum sensing period.

It is clear that under the H_0 hypothesis, $z_i(k) = n_i(k) \sim \mathcal{N}(0, \sigma_n^2)$, thus, y_i will follow a central Chi-squared distribution with K degrees of freedom:

$$y_i = \sum_{k=1}^{K} \left[\frac{n_i(k)}{\sigma_n}\right]^2 = \sum_{k=1}^{K} \hat{z}_i^2(k) \text{ where } \hat{z}_i(k) \sim \mathcal{N}(0,1)$$
$$\therefore \quad y_i \sim \mathcal{X}_K^2 \tag{4}$$

On the other hand, under hypothesis H_1, the signal at the i_{th} SU will be the composition of the PU signal scaled by the channel gain plus the receiver Gaussian noise. If we assume the transmitted PU signal as zero-mean Gaussian random variable with variance σ_s^2, then the *estimated normalized energy* level will follow a Gamma distribution with shape $\frac{K}{2}$ and scale $2(1+\gamma_i)$:

$$y_i = \sum_{k=1}^{K} \left[\frac{h_i x(k) + n_i(k)}{\sigma_n}\right]^2 = \sum_{k=1}^{K} \hat{z}_i^2(k) \tag{5}$$

where $\hat{z}_i(k) \sim \mathcal{N}\left(0, 1 + \frac{h_i^2 \sigma_s^2}{\sigma_n^2}\right) \; \therefore y_i \sim \Gamma\left(\frac{K}{2}, 2(1+\gamma_i)\right)$

where γ_i is the signal-to-noise ratio given by: $\gamma_i = \left(\frac{h_i \sigma_s}{\sigma_n}\right)^2$

On the *conventional energy detection* scheme, each SU infers the spectrum occupancy status by comparing the normalized sensed energy level to a given threshold λ:

$$s_i = \begin{cases} H_1, & \text{if } y_i \geq \lambda \\ H_0, & \text{if } y_i < \lambda \end{cases} \tag{6}$$

The probability of false alarm is defined as $P_{\text{fa}} = P(y \geq \lambda | H_0)$. Likewise, the probability of detection is defined as $P_{\text{d}} = P(y \geq \lambda | H_1)$. Hence, given the statistics of hypothesis H_0, we can write P_{fa} for the i_{th} SU as the right-tail probability of a central Chi-squared random variable:

$$P_{\text{fa}_i} = \int_{\lambda}^{\infty} f(y_i|H_0) dy \triangleq Q_{\mathcal{X}_K^2}(\lambda) \tag{7}$$

where $f(y_i|H_j)$ is the conditional probability density function (PDF) for the estimated normalized energy level at the the i_{th} SU, given hypothesis H_j, with $j = 0, 1$.

In turn, the threshold parameter λ can be obtained from (7) by fixing a target false alarm probability (P_{fa}^*) as:

$$\lambda = Q_{\mathcal{X}_K^2}^{-1}(P_{\text{fa}}^*) \tag{8}$$

III. Soft Combination Centralized Cooperative Spectrum Sensing

In centralized cooperative spectrum sensing, the term soft combination refers to schemes where the estimated energy level on the SU is transmitted for further processing in the fusion center, which will infer the channel status and transmit the decision back to the secondary users, concluding the sensing time period. Within this category, we shall enumerate and briefly analyze three Bayesian methods and the well-established maximum ratio combining method.

A. Maximum Ratio Combining (MRC)

The Maximum Ratio Combining (MRC) technique estimates the channel status by combining the weighted energy levels obtained at each SU and communicated with the fusion center.

$$\hat{S} = \begin{cases} H_1, & \text{if } \sum_{i=1}^{N} w_i y_i \geq \lambda \\ H_0, & \text{otherwise} \end{cases} \tag{9}$$

where w_i is the weight for the i_{th} SU energy level, defined as the normalized SNR over N cooperate nodes, seen on each SU, i.e., $w_i = \frac{\gamma_i}{\sum_N \gamma_i}$ and λ is given by Eq. (8).

This technique achieves optimum performance at the cost of estimating and transmitting the SNR and exact energy level of each SU to a fusion center.

B. Weighted Bayesian (WB) Estimator

The Weighted Bayesian estimator (WB) takes advantage of the knowledge on the statistics of the estimated energy level and computes the *a posteriori* probability of occurrence for each hypothesis, given the energy level estimated on each SU. The estimator then combines the calculated probabilities of each SU weighted by their respective average SNR $\bar{\gamma}_i$, hence:

$$P(H_1|y) = \frac{1}{\sum_{i=1}^{N} \bar{\gamma}_i} \sum_{i=1}^{N} \bar{\gamma}_i P(H_1|y_i) \tag{10}$$

where $P(H_1|y_i)$ is given by Bayes Theorem:

$$P(H_1|y_i) = \frac{P(y_i|H_1)P(H_1)}{P(y_i|H_0)P(H_0) + P(y_i|H_1)P(H_1)} \tag{11}$$

where $P(H_1)$ and $P(H_0)$ are the PU transmission *a priori* probability and its complement, respectively, discussed in Section IV.

From (4) and (6), Eq. (11) can be rewritten as:

$$P(H_1|y_i) = \frac{f_{\Gamma_{\frac{K}{2}}}(y_i)P(H_1)}{f_{\mathcal{X}_N^2}(y_i)P(H_0) + f_{\Gamma_{\frac{K}{2}}}(y_i)P(H_1)} \tag{12}$$

In order to reduce the overhead, the conditional probability $P(H_1|y_i)$ can be calculated on each SU and transmitted to a fusion center instead of the estimated energy level, alongside its respective average SNR $\bar{\gamma}_i$.

Finally, the WB estimator infers the channel status (occupied or empty) based on the following rule:

$$\hat{S} = \begin{cases} H_1, & \text{if } P(H_1|\mathbf{y}) \geq 1 - P_{\text{fa}}^* \\ H_0, & \text{otherwise} \end{cases} \tag{13}$$

where $\mathbf{y} = [y_1, y_2, \ldots y_N]$ is the estimated energy level vector.

C. Learning-Based Naive Bayes (NB) Classifier

The Naive Bayes (NB) classifier makes a simplifying assumption regarding the *estimated energy* level on each SU (y_i), by considering that y_i, for $i = 1$ to N, are mutually independent, thus $f(\mathbf{y}|H_j) = \prod_{i=1}^{N} f(y_i|H_j)$, with $j = \{0, 1\}$. Furthermore, by considering that $f(y_i|H_j)$ is Gaussian, $f(\mathbf{y}|H_j)$ becomes a multivariate Gaussian distribution with diagonal covariance matrix.

Indeed, as the number of samples K increases, (4) can be well approximated by a Gaussian distribution with mean K and variance $2K$. Likewise, (6) can be well approximated by a Gaussian distribution with mean $K(1 + \gamma_i)$ and variance $2K(1 + \gamma_i)^2$.

With this in mind, we can model $f(\mathbf{y}|H_j)$ using a *machine learning* approach. Considering a training set with samples of y_i of size Ω_{train}, in which the channel status is known, the Naive Bayes model can estimate the probability density functions parameters of each hypothesis as:

$$\begin{aligned} \hat{\mu}_{i,H_{1,0}} &= \frac{1}{\Omega_{\text{train}}^{H_{1,0}}} \sum_{\omega=1}^{\Omega_{\text{train}}^{H_{1,0}}} y_{i,H_{1,0}}[\omega] \\ \hat{\sigma}_{i,H_{1,0}}^2 &= \frac{1}{\Omega_{\text{train}}^{H_{1,0}} - 1} \sum_{\omega=1}^{\Omega_{\text{train}}^{H_{1,0}}} (y_{i,H_{1,0}}[\omega] - \hat{\mu}_{i,H_{1,0}})^2 \end{aligned} \tag{14}$$

where $\Omega_{\text{train}}^{H_{1,0}}$ refers to the section of the training set which comprises hypothesis H_1 or H_0.

After the training phase, the Naive Bayes classifier can estimate the channel occupancy probability given the detected energy $\mathbf{y}$ through Bayes Theorem, Eq. (11), where $P(H_1)$ and $P(H_0)$ are the *a priori* probabilities of each hypothesis, estimated in the training phase as the number of occurrences of each hypothesis divided by the number of training samples.

Finally, the channel status is inferred based on $P(H_1|\mathbf{y})$ and the target false alarm probability in a same way as obtained in the WB estimator, Eq. (13). However, differently of WB estimator, by using this NB learning-based approach, the secondary users are not required to estimate the average signal-to-noise ratio $\overline{\gamma}_i$, as it can be inferred from the training phase with the labeled samples, Eq. (14).

D. Gaussian Mixture Model

The Gaussian Mixture (GM) estimator assumes that the estimated energy levels given each hypothesis form a multivariate Gaussian distribution, which, according to the central limit theorem, is a valid approximation for large number of samples K. For simplification, we further assume that the distributions are mutually independent.

Thus, the GM model applied to spectrum sensing problem can be fully specified by the mean matrix $\boldsymbol{M}$, variance matrix $\boldsymbol{\Sigma}^2$ and mixing proportions vector $\boldsymbol{\pi}$. From (4) and (6), we have:

$$\begin{aligned} \boldsymbol{M} &= \begin{bmatrix} K & \ldots & K \\ K(1 + \bar{\gamma}_i) & \ldots & K(1 + \bar{\gamma}_N) \end{bmatrix} \\ \boldsymbol{\Sigma}^2 &= \begin{bmatrix} 2K & \ldots & 2K \\ 2K(1 + \bar{\gamma}_i)^2 & \ldots & 2K(1 + \bar{\gamma}_N)^2 \end{bmatrix} \\ \boldsymbol{\pi} &= \begin{bmatrix} P(H_0) & P(H_1) \end{bmatrix}^T \end{aligned} \tag{15}$$

Given the GM model, the estimator calculates the *a posteriori* probability $P(H_1|\mathbf{y})$ as in Eq. (11) and infers the channel occupancy by applying Eq. (13). Therefore, the GM model performs as an asymptotic version of the Naive Bayes classifier discussed in III-C and can be seen as a upper-bound on the performance of NB as the size of the training set increases.

IV. Numerical Results

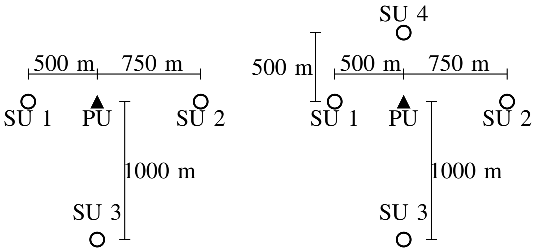
In order to compare the performance of the analyzed Bayesian techniques, we carried out Monte Carlo simulations with 5×10^4 realizations, considering two CRN scenarios. For both scenarios, we considered the parameters described in Table I. In addition to the soft combination techniques, we considered two well-established hard combination methods, where the secondary users decisions based on Eq. (6) are transmitted to the fusion center, namely the AND rule and OR rule, for comparison purpose.

Table I
Adopted system parameters for the Monte Carlo simulations.

Bandwidth (w)	5 MHz	Sampling frequency (f_s)	10 MHz
Noise PSD (η_0)	-153 dBm	PU active prob. ($P(H_1)$)	0.5
PU TX power (σ_s^2)	0.1 mW	Number of SUs (N)	3 and 4
Sensing time (τ)	5 μs	Number of samples (K)	50
Training set size (Ω_{train})		$\Omega_{\text{train}}^{H_1} + \Omega_{\text{train}}^{H_0} = 500$	

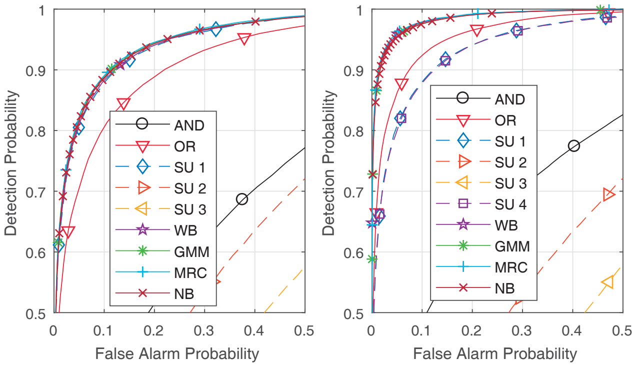
A. Scenario I

As depicted in Fig. 1a, the first scenario considers a diverse setup with three secondary users, with various distances from the PU, thus, various SNR levels can be emulated. Hence, we have evaluated the receiver operating characteristic (ROC) curve for each technique discussed, both cooperative and individual spectrum sensing strategies operating under scenario 1. The average SNR obtained on each SU are as follows: $\bar{\gamma}_1 = -2$ dB, $\bar{\gamma}_2 = -9$ dB and $\bar{\gamma}_3 = -14$ dB. From Fig. 2a we notice that all the soft-combination methods evaluated performed really closely. Indeed, from Tab. II, the values for the *area under the ROC curve* (AuC) demonstrate that although the performances are similar, the Weighted Bayesian estimator is slightly outperformed by the simpler Gaussian Mixture model and the learning-based Naive Bayes classifier. Besides, all the soft-combination methods are quite superior in terms of ROC performance regarding both hard combination cooperative SS techniques, namely rule AND and OR.



(a) CRN scenario I (b) CRN scenario II

Figure 1. CRN simulation scenarios.



(a) ROC curve for Scenario I (b) ROC curve for Scenario II

Figure 2. ROC curves for scenarios I and II.

In order to better assess the effect of the training set size on the performance of the NB classifier, we evaluated six different training set sizes and compared the performance in terms of the ROC curve and AuC measures, as well as the Gaussian Mixture model, which acts as the upper-bound. Figure 3 summarizes our finding. It is clear that for training sets larger than 250 samples the NB performance is sufficiently close to the upper-bound (GM model), with marginal improvements for increases in the training set when $\Omega_{\text{train}} > 1000$ samples.

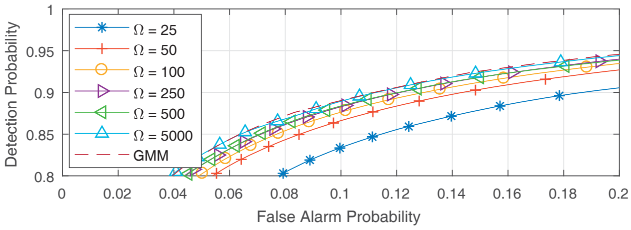


Figure 3. Naive Bayes performance for various training set sizes.

B. Scenario II

Next, we consider Scenario II depicted in Fig. 1b. As an extension to Scenario I, we consider the addition of another SU at 500 m from the primary user, in order to evaluate the cooperative performance benefits.

The addition of another SU with an average SNR of $\bar{\gamma}_4 = -2$ dB incurs in an increase in the detection probability for all cooperative methods evaluated, as depicted in Fig. 2b, improving substantially the overall CRN spectrum sensing performance. In contrast to the ROC curve for Scenario I, Fig. 2b corroborates that the centralized cooperative techniques analyzed herein remarkably outperform all individual SUs, demonstrating that these methods can leverage the addition of more secondary users to the cooperative schemes, at the cost of exchanging information between the cooperative SU nodes and the fusion center, reducing the spectral efficiency of the CRN.

Table II
MONTE CARLO SIMULATION RESULTS (AUC).

SCENARIO	MRC	GM	WB	NB	AND	OR
Scenario I	0.9627	0.9615	0.9603	0.9608	0.7104	0.9314
Scenario II	0.9913	0.9902	0.9909	0.9891	0.7735	0.9716

V. DISCUSSION AND FINAL REMARKS

In this paper, we compared cooperative spectrum sensing techniques of Bayesian nature such as a weighted Bayesian estimator, a Gaussian mixture estimator and the learning-based Naive Bayes classifier to the well-established centralized techniques such as the Maximum Ratio Combining, rule AND and rule OR techniques.

As an alternative for the high overhead of the Maximum Ratio Combining method, we presented the weighted Bayesian estimator, which transmits the channel occupancy probability to the fusion center, instead of the exact estimated energy level.

For systems that can not afford to estimate the SNR on the SU nodes, the Learning-Based Naive Bayes classifier can be seen as an attractive alternative. In order to infer the channel status, the NB requires a prior training phase with estimated energy levels of each SU and the respective channel status. It was shown that for a sufficiently large training set, the NB performs similarly as the Gaussian mixture model technique with mutually independent distributions.

Based on Monte Carlo simulations, we were able to demonstrate the performance of each SS method operating under two practical cognitive radio scenarios. The numerical results allow infer that all soft-combination cooperative methods performed closely for both scenarios.

REFERENCES

[1] Y. Chen and H.-S. Oh, "A survey of measurement-based spectrum occupancy modeling for cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 848–859, October 2014.

[2] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, August 1999.

[3] F. hsun Tseng, L. der Chou, H. chieh Chao, and J. Wang, "Ultra-dense small cell planning using cognitive radio network toward 5g," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 76–83, December 2015.

[4] Z. Li, F. R. Yu, and M. Huang, "A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 383–393, September 2009.

[5] J. Ma, G. Zhao, and Y. Li, "Soft combination and detection for cooperative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4502–4507, December 2008.

[6] 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, October 2013.

[7] F. Azmat, Y. Chen, and N. Stocks, "Analysis of spectrum occupancy using machine learning algorithms," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 6853–6860, October 2015.

A.2 Machine Learning Models for Cooperative Spectrum Sensing

Title:	*Machine Learning Models for Cooperative Spectrum Sensing*
Authors:	Caio Henrique Azolini Tavares, José Carlos Marinello Filho Mario Lemes Proença Jr. and Taufik Abrão
Submitted:	September/2019
Journal:	IET Communications

IET Communications

Machine Learning-based Models for Spectrum Sensing in Cooperative Radio Networks

COM-2019-0941 | Special Issue: Cognitive and AI-enabled Wireless and Mobile Communications

Submitted by: Caio Tavares, José Carlos Marinello, Mario Lemds Proenca Junior, TAUFIK ABRAO

Keywords: COGNITIVE RADIO NETWORKS, SPECTRUM SENSING, LEARNING (ARTIFICIAL INTELLIGENCE), COMPUTATIONAL COMPLEXITY, PERFORMANCE ANALYSIS

Machine Learning-based Models for Spectrum Sensing in Cooperative Radio Networks

Caio Henrique Azolini Tavares, Jose Carlos Marinello, Mario Lemes Proenca Jr.
Taufik Abrao

Abstract—**In this paper, we consider the application of machine learning (ML) models in cooperative spectrum sensing (SS) of cognitive radio networks (CRNs). Based on statistical analysis of classic energy detection (ED) scheme, the probability of detection and false alarm are derived, which depends solely on the number of samples and signal-to-noise ratio of the SUs. The channel occupancy detection obtained from established analytical techniques such as MRC and AND/OR rules are compared to different machine learning techniques, including Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Naive Bayes (NB), based on receiver operating characteristic (ROC) and area under the curve (AuC) metrics. By using standard profiling tools, we obtain the computational performance of the analyzed models during the training phase, a critical step for operating in CRNs. Ultimately, the results demonstrate that the MLP machine learning technique presents a better trade-off between training time and channel detection performance.**
***Keywords* – Cognitive Radios (CR); Spectrum Sensing (SS); Machine Learning (ML) techniques; Computational Performance;**

I. Introduction

Cognitive radios (CR) systems are a proposed solution to the spectrum scarcity problem in radio frequency (RF) environment that aims to improve the overall spectrum utilization. Several studies showed [1]–[3] that licensed spectrum bands are often not occupied by the licensed users, thus creating the opportunity for other devices to access the unoccupied spectrum opportunistically. These opportunistic devices denoted as secondary users (SU), need to be able to sense the spectrum to assess the presence or absence of licensed users, denoted as primary users (PU), either individually or cooperatively. The idea of cognitive radios was first introduced by Joseph Mitola III in 1999 [4], but has been given much attention recently due to the proposed heterogeneous nature of 5G networks [5]–[8].

C. Tavares, J. C. Marinello and T. Abrao are with the Department of Electrical Engineering, State University of Londrina, Parana, Brazil. E-mail: caio.tavares11@gmail.com; jcmarinello@uel.br; taufik@uel.br

M. L. Proenca Jr. is with the Department of Computer Science, State University of Londrina, Parana, Brazil. E-mail: proenca@uel.br.

Spectrum sensing in cognitive radios (SS-CRs) still poses a challenge for high-performance and high-energy efficient systems since SS performance is often proportional to the spectrum sensing period. In turn, SS-CRs is an energy consuming task that also degrades the spectral efficiency of the secondary users, since they need to spend time and energy on a task that does not result in transmitted bits.

Machine learning (ML) has received increasing interest and found application in many fields recently. ML is a way of programming computers to optimize a performance criterion using example data or experience [9]. Such interest is due to the its ability to apply complex calculations to evaluate and interpret patterns and to its ability to interpret patterns and structures in data, enabling learning, reasoning, and decision making. This apparent self-learning characteristic associated to ML techniques is by itself mostly based on applied statistics, whereas the training and inference capabilities owe their efficiency to great computer science algorithms.

There are several networking open problems been treated under the ML perspective, including a) traffic prediction; b) traffic classification; c) traffic routing; d) congestion control, including important issues such as queue management, congestion inference and packet loss classification; e) resource management, which comprises admission control and resource allocation policies; f) fault management; g) QoS and QoE management; h) network security, aggregating anomaly and intrusion detection; among others. A comprehensive survey on machine learning applied to networking is discussed in [10].

Specifically, in the context of cognitive radio networks (CRNs), several research papers related to machine learning for spectrum sensing have been published. These ML-based sensing techniques aim at detecting the availability of frequency channels by formulating the process as a classification problem in which the classifier, supervised or unsupervised, has to decide between two states of each frequency channel: free or occupied.

Moreover, cooperative SS-CRNs have been performed by ML. The existing works can be classified into two main categories. The technique in the first category uses two steps. In the first step, unsupervised machine learning techniques are used to analyze data and discover the primary user's patterns. In the second step, supervised machine learning techniques are used to train the model with the data labeled in the first step [11]. For instance, a two-step machine learning model for spectrum sensing can be constructed. In the first step, for instance, the K-means algorithm could be used to identify the state of the primary user's presence. In the second step, support vector machine (SVM) or other types of classifiers can be used to attribute the new input data into one of the classes specified by the K-means method used in the first step. Techniques of the second category assume that the classes are known, and they are based on supervised machine learning to train models. In the current work, we follow the second category approach. For example, existing works in the literature that use only one step in which supervised machine-learning classifiers, such as K-nearest neighbor, support vector machine, Naive Bayes, and decision tree, are applied [11].

Since the task of determining the channel status based on spectrum sensing is due to its nature a classification task, several authors have considered the use of ML models as inference tools. In [12], the authors propose and compare the performance of several supervised and unsupervised ML techniques for cooperative SS purpose, such as Support Vector Machines (SVM), the K-means clustering algorithm and Gaussian Mixture Model (GMM), but do not provide a comparison of detection performance over different scenarios of interest, such as considering distinct training set sizes or different channel scenarios of practical interest. In [13], the authors study the use of ML algorithms for spectrum occupancy in cognitive radio networks (CRNs), which include the Naive Bayesian Classifier (NBC). In [14], the authors propose user grouping algorithms to improve spectrum sensing results and SVM training time and in [15] the authors enumerate the pros and cons of several unsupervised and supervised machine learning techniques applied to spectrum sensing, such as the requirement of data labeling for supervised models and the risk of overfitting. Authors in [16] proposed a centralized SS-CRN scheme based on K-nearest neighbor. In the training phase, each CR user produces a sensing report under varying conditions and, based on a global decision, either transmits or stays silent. The local decisions of CR users are combined through a *majority voting* at the fusion center and a global decision is returned to each CR user, implying in a spectral overhead.

SVM-based cooperative SS model with user grouping method is discussed in [17]. User grouping procedures reduce cooperation overhead and effectively improve detection performance. Hence, users in CRN are grouped before the cooperative sensing process using energy data samples and a properly ML model. Authors compare three grouping methods, the first divides normal and abnormal users into two groups, while the second grouping algorithm distinguishes redundant and non-redundant users, and the third grouping algorithm selects users within a subset that minimising average correlation. The performances of the three algorithms were quantified in terms of the average training time, classification speed and classification accuracy.

Finally, in [18], a low-dimensional probability vector is proposed as the feature vector for machine learning based classification, instead of the high-dimensional energy vector in a CRN with a single primary user and N secondary users. Such method down-converts a high-dimensional feature vector to a constant two-dimensional feature vector for ML techniques while keeping the same spectrum sensing performance. Due to its lower dimension, the probability vector-based classification requires a smaller training duration and a shorter classification time.

A. Contributions

This work considers the application of supervised ML models to the task of channel status inference based on cooperative energy detection spectrum sensing. By considering the *a posteriori* probability of channel occupancy, we aim to obtain a clear tradeoff characterization between computational complexity and classification performance for each model when compared to traditional analytical cooperative SS-CRN techniques, while also raising relevant issues on the system implementability.

B. Paper Organization

The remainder of this paper is organized as follows. Section II discusses the system model for the CRN and the statistics of the primary user signal for energy detectors. Section III includes the modeling of well known analytical models for cooperative spectrum sensing. Section IV explores the application of supervised machine learning techniques. Section V shows comparative results of the techniques through Monte Carlo simulations. Finally, Section VI offers remarks and main conclusions.

II. System Model

We consider a cognitive radio network (CRN) with one PU and n SUs, where each secondary user employs energy detection for spectrum sensing with sensing period τ and sensing bandwidth w. The sampling frequency is considered to be at the Nyquist rate, $f_s = 2w$. Thus the SU acquires $K = 2w\tau$ samples at each sensing period.

We can write the signal at the i_{th} SU according to two hypotheses:

$$\begin{cases} H_1 : z_i(k) = h_i x(k) + n_i(k), \text{ if the PU is active} \\ H_0 : z_i(k) = n_i(k), \text{ otherwise} \end{cases} \tag{1}$$

where h_i is the channel coefficient from the PU to the i_{th} SU, $x(k)$ is the transmitted PU signal, and $n_i(k)$ is the noise at the i_{th} SU receiver considered to be a zero-mean Gaussian random variable with variance σ_n^2.

The channel coefficient h_i is described by the path-loss and fading components:

$$h_i = g_i D_i^{-\frac{\alpha}{2}} \tag{2}$$

where g_i is the fading component, D_i is the Euclidean distance between the PU and the i_{th} SU and α is the path-loss exponent, which we consider as 4, indicating a non-line-of-sight (NLOS) channel environment.

At the end of the spectrum sensing period, the estimated normalized energy level at the i_{th} SU is given by:

$$y_i = \frac{1}{\sigma_n^2} \sum_{k=1}^{K} z_i(k)^2 \tag{3}$$

Under hypothesis H_0, $z_i(k) = n_i(k) \sim \mathcal{N}(0, \sigma_n^2)$, thus, y_i will follow a central Chi-squared distribution with K degrees of freedom:

$$\begin{gathered} y_i = \sum_{k=1}^{K} \left(\frac{n_i(k)}{\sigma_n} \right)^2 = \sum_{k=1}^{K} \hat{z}_i(k)^2 \text{ where } \hat{z}_i(k) \sim \mathcal{N}(0,1) \\ \therefore\ y_i \sim \mathcal{X}_K^2 \end{gathered} \tag{4}$$

On the other hand, under hypothesis H_1, the signal at the i_{th} SU will be the composition of the PU signal scaled by the channel gain plus the Gaussian receiver noise. If we assume the transmitted PU signal as zero-mean Gaussian random variable with variance σ_s^2, then the *estimated energy level* will follow a Gamma distribution with shape $\frac{K}{2}$ and scale $2(1+\gamma_i)$:

$$\begin{aligned} y_i &= \sum_{k=1}^{K} \left(\frac{h_i x(k) + n_i(k)}{\sigma_n} \right)^2 \\ &= \sum_{k=1}^{K} \hat{z}_i(k)^2 \text{ , where } \hat{z}_i(k) \sim \mathcal{N}\left(0, 1 + \frac{h_i^2 \sigma_s^2}{\sigma_n^2}\right) \\ &\therefore y_i \sim \Gamma\left(\frac{K}{2}, 2(1+\gamma_i)\right) \end{aligned} \tag{5}$$

where γ_i is the signal-to-noise (SNR) ratio given by: $\gamma_i = \left(\frac{h_i \sigma_s}{\sigma_n}\right)^2$

On the conventional energy detection scheme, each SU infers the spectrum occupancy status by comparing the normalized sensed energy level to a given threshold λ:

$$s_i = \begin{cases} H_1, & \text{if } y_i \geq \lambda \\ H_0, & \text{if } y_i < \lambda \end{cases} \tag{6}$$

The probability of false alarm is defined as $P_{\text{fa}} = P(y \geq \lambda | H_0)$. Likewise, the probability of detection is defined as $P_d = P(y \geq \lambda | H_1)$. Hence, given the statistics of hypothesis H_0, we can write P_{fa} for the i_{th} SU as the right-tail probability of a central Chi-squared random variable:

$$P_{\text{fa}_i} = \int_{\lambda}^{\infty} f(y_i | H_0) dy \triangleq Q_{\mathcal{X}_N^2}(\lambda) \tag{7}$$

In turn, the threshold parameter λ can be obtained from (7) by fixing a target false alarm probability (P_{fa}^*) as:

$$\lambda = Q_{\mathcal{X}_N^2}^{-1}(P_{\text{fa}}^*) \tag{8}$$

Using the definition of the incomplete Gamma function, Eq. (8) can be rewritten as [19]:

$$\lambda = 2\Gamma_u^{-1}\left(P_{\text{fa}}^*, \frac{K}{2}\right) \tag{9}$$

where $\Gamma_u(x, n)$ is the upper incomplete Gamma function, defined as:

$$\Gamma_u(x, n) = \frac{1}{\Gamma(x)} \int_{n}^{\infty} t^{x-1} e^{-t} dt \tag{10}$$

and $\Gamma(x)$ is the Gamma function.

Likewise, given λ we can calculate the probability of detection offered by the i_{th} SU as the right-tail probability of a Gamma distribution with shape $\frac{K}{2}$ and scale $2(1+\gamma)$:

$$P_{d_i} = \int_{\lambda}^{\infty} f(y_i | H_1) dy \triangleq Q_{\Gamma}\left(\lambda; \frac{K}{2}, 2(1+\gamma_i)\right) \tag{11}$$

which, in turn, can be rewritten as:

$$P_{d_i} = \Gamma_u\left(\frac{\lambda}{2(1+\gamma_i)}, \frac{K}{2}\right) \tag{12}$$

III. Conventional SS-CRN Techniques

So far, the discussed methodology for the i_{th} SU to assess the presence or absence of a primary user involves only the estimated energy level on the i_{th} SU. Now we shall enumerate and briefly analyze classical analytical methods of cooperative spectrum sensing, where the estimated energy level on all secondary users are transmitted through a service layer to a fusion center, which will apply deterministic decision rules aiming at deciding cooperatively whether a PU is present or not by combining the SU individual decisions. In the following the three main deterministic decision rules for SS cooperative networks, namely AND, OR and MRC rules are revisited.

A. AND Rule

In the AND rule, the fusion center decides the occupancy state of the channel by comparing the sensed energy level of each SU to the threshold λ from Eq. (9) and then applying the AND logical operation on the result, as shown below:

$$\hat{S} = \begin{cases} 1, & \text{if } (\hat{s}_1 \odot \hat{s}_2 \odot \ldots \hat{s}_N) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\odot$ denotes the logical AND operator.

B. OR Rule

The OR rule is similar to the AND rule, in which it compares the energy level from each SU to λ and then applies the logical OR operation on the result:

$$\hat{S} = \begin{cases} 1, & \text{if } (\hat{s}_1 \oplus \hat{s}_2 \oplus \ldots \hat{s}_N) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $\oplus$ denotes the logical OR operator.

C. Maximum Ratio Combining

The maximum ratio combining (MRC) technique estimates the channel status by combining the weighted energy levels obtained at each SU and communicated with the fusion center.

$$\hat{S} = \begin{cases} H_1, & \text{if } \sum_{i=1}^{N} w_i y_i \geq \lambda \\ H_0, & \text{otherwise} \end{cases} \tag{15}$$

where w_i is the weight for the i_{th} SU energy level, defined as the normalized average SNR over N cooperate nodes $\bar{\gamma}_i$, seen on each SU, i.e., $w_i = \frac{\bar{\gamma}_i}{\sum_N \bar{\gamma}_i}$ and λ is given by Eq. (8).

This technique achieves optimum performance at the cost of increasing complexity and overall spectral efficiency reduction, due to the requirement of estimating and transmitting the SNR and exact energy level of each SU to a fusion center (centralized decion).

IV. Machine Learning Techniques

The primary motivation for using ML models in spectrum sensing is the ability to operate in the absence of knowledge of the CR network parameters, such as the signal-to-noise ratio γ_i in SUs and the *a priori* hypothesis probabilities $P(H_0)$ and $P(H_1)$. The only requirement in the case of supervised learning methods is a set of labeled energy samples in order to train the models. In practical terms, this means that the primary users need to cooperate with the training phase of the secondary users by providing the channel status regularly. As shown, in Fig. 1 a simple CRN configuration with $N = 2$ SUs, one PU and one fusion center is sketched.

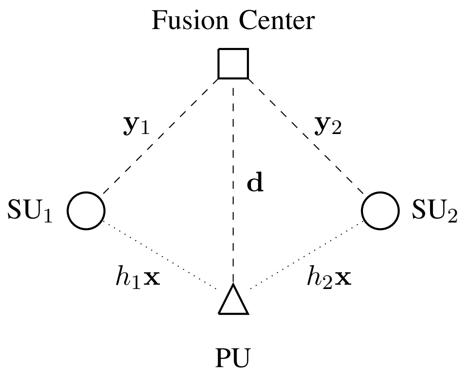


Figure 1. Example of a cognitive radio scenario training phase. The SUs provide the fusion center with $\mathbf{y} \in \mathbb{R}^{M\times 1}$ energy samples, while the PU provide the corresponding channel status vector $\mathbf{d}$.

The general objective of a supervised machine learning model is to devise a system that, given a set of training inputs and the corresponding labelled output, such system is able to infer on the output of an unseen example or scenario with a certain level of confidence.

In the context of channel status inference, this means that given a set $\mathbf{Y}_{\text{train}} \in \mathbb{R}^{M\times N}$ of M energy samples detected by N cooperative SUs and a corresponding set $\mathbf{d} \in \mathbb{R}^{M\times 1}$ where $d_m \in \{0;\ 1\}$ of channel status, we want to obtain a hypothesis function $h(\mathbf{y}) : \mathbb{R}^N \to \mathbb{R}$ which can estimate the channel status $\hat{S}$, i.e., for a given $\mathbf{y}_m$, we want to be able to predict d_m. For clarification purpose, Fig. 2 shows the received energy level on 3 SUs and the corresponding channel status for $M = 10^4$ samples. Clearly, the classes are non-separable in input space.

Concretely, the model training procedure can be seen as obtaining a *hypothesis function* such as the linear:

$$h(\mathbf{y}) = \mathbf{y}\mathbf{w} + b \tag{16}$$

where $\mathbf{w} \in \mathbb{R}^{N\times 1}$ optimally separates the two classes or channel status by minimizing an error function

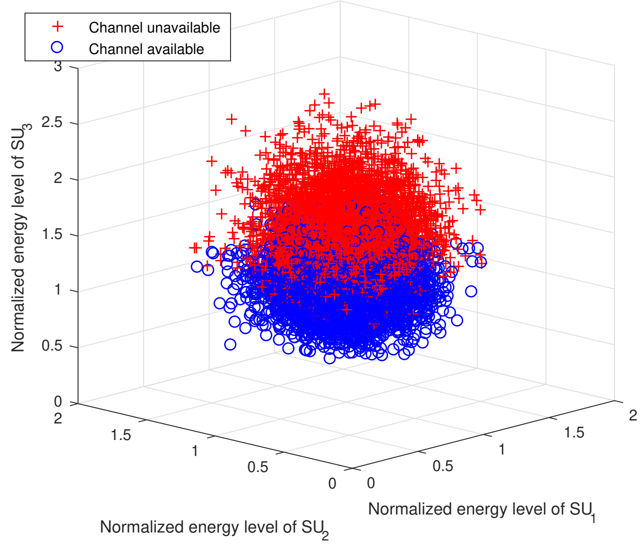


Figure 2. Example of non-separable $\mathbb{R}^3$ input space. Each dimension is the received energy level on a secondary user.

$\mathrm{E}(h(\mathbf{Y}_{\text{train}}), \mathbf{d})$. The decision region of (16) can also be interpreted as the hyperplane $h(\mathbf{y}) = 0$. With this in mind, we can write down the general goal of a supervised machine learning problem:

$$\underset{\mathbf{w},b}{\text{minimize}} \ \frac{1}{M} \sum_{i=1}^{M} \mathrm{E}(h(\mathbf{y}_i), d_i) \tag{17}$$

where $\mathbf{w}$ and b are the parameters of the hypothesis function in Eq. (16), and $\mathrm{E}(\cdot)$ is the *error function*.

In simple terms, most supervised ML models differ only in three aspects:

a) implementation of the error function $\mathrm{E}(\cdot)$;
b) formulation of the hypothesis function $h(\mathbf{y})$;
c) optimization methodology applied on $\mathrm{E}(\cdot)$.

A. Multilayer Perceptron (MLP)

Neural networks are known as universal function approximators and perhaps the most well-known machine learning technique. The neural network maps the input vector $\mathbf{y}$ to the output $\mathbf{o}$ through a set of weighted nonlinear functions. Given its relative simplicity and success in the context of pattern recognition [20], we consider a multilayer perceptron with one hidden layer, which is based on Rosenblatt perceptron developed in [21].

In the forward phase, an input signal flows from left to right of the network. The signal value at the input of any given neuron j at layer ℓ can be written as:

$$a_j^{(\ell)} = \sum_i w_{ij}^{(\ell)} o_i^{(\ell-1)} \tag{18}$$

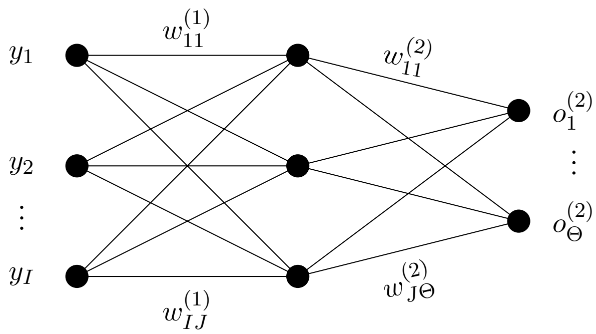


Figure 3. General representation of a feed-forward neural network with one hidden layer.

where $o_i^0 = y_i$

Similarly, the output of any neuron j is given by:

$$o_j^{(\ell)} = \sigma\left(a_j^{(\ell)}\right) \tag{19}$$

where $\sigma(\cdot)$ is the activation function, which has only two requirements: to be differentiable and nonlinear. Often chosen to be the logistic sigmoid function or hyperbolic tangent. In this paper, we chose the former:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{20}$$

Being specific, we consider a neural network with one hidden layer, one output unit, i.e. $\Theta = 1$ (since we are interested in binary classification) and with the number of inputs equal to the number of SUs, i.e. $I = N$. Also, we consider the number of neurons of the hidden layer to be equal to the number of inputs N, so $J = N$. Therefore, we can write the output of any neuron in the hidden layer as:

$$o_j^{(1)} = \sigma\left(\sum_{i=0}^{N} w_{ij}^{(1)} y_i\right) \tag{21}$$

and similarly for the output neurons:

$$o_\theta^{(2)} = \sigma\left(\sum_{j=0}^{N} w_{j\theta}^{(2)} o_j^{(1)}\right) \tag{22}$$

in both cases, y_0 and $o_0^{(1)}$ are known as bias inputs and are equal to 1. They are necessary in order to shift the activation function away from the origin.

By considering a training vector of desired binary outputs $\mathbf{d} \in \mathbb{R}^{M\times 1}$ with $d_m \in \{0; 1\}$ for M input vectors $\mathbf{y}$, we can interpret the output o of the neural network (22) as the conditional probability $p(H_1|\mathbf{y})$, whereas the probability $p(H_0|\mathbf{y})$ is given by $1 - o$. With this in mind, we can write the conditional distribution of desired outputs as a Bernoulli distribution: [20]

$$p(d|\mathbf{y}, \mathbf{w}) = o^d \left(1 - o\right)^{1-d} \tag{23}$$

where the set of all weights and biases have been combined into vector $\mathbf{w}$.

By taking the negative log likelihood of (23) and, by considering that the training set is composed of independent observations, the error function becomes the *cross-entropy error function* for the input vector $\mathbf{y}_m$ [20]:

$$\begin{aligned} \mathrm{E}(\mathbf{w}) &= -\sum_{m=1}^{M} E_m(\mathbf{w}) \\ &= -\sum_{m=1}^{M} \{d_m \ln o_m + (1-d_m)\ln(1-o_m)\} \end{aligned} \tag{24}$$

Since it is not possible to arrive at the optimum weight vector analytically [20], we can resort to a numerical approach such as stochastic gradient descent (25) in order to find $\mathbf{w}$ that minimizes (24):

$$\mathbf{w}^{[\tau+1]} = \mathbf{w}^{[\tau]} - \eta \nabla \mathrm{E}_m(\mathbf{w}^{[\tau]}) \tag{25}$$

where $\eta > 0$ is known as the *learning rate* parameter and controls the step size of the update to the weight vector.

A prevalent method for obtaining the error function gradient on each neuron is known as backpropagation. Putting simply, we propagate an error signal from the output nodes to the hidden nodes and apply a penalty to the associated weight according to the cost function. For a network with Θ outputs and two layers (same as depicted in Fig. 3), the error on the output node θ is trivial:

$$\delta_\theta = o_\theta - d_\theta \tag{26}$$

whereas on the j_{th} hidden neuron:

$$\delta_j = \sigma'\left(a_j^{(1)}\right)\sum_\theta w_{\theta j}^{(2)}\delta_\theta \tag{27}$$

where $\sigma'(\cdot)$ is the derivative of the activation function $\sigma(\cdot)$.

Once we have the errors for every neuron, we can easily obtain the derivative:

$$\frac{\partial \mathrm{E}_m}{\partial w_{ij}} = \delta_j o_i \tag{28}$$

Finally, once the neural network is trained, the channel status inference for an unseen example is made based on:

$$\hat{S}_{\text{MLP}} = \begin{cases} H_1, & \text{if } o \geq 1 - P_{\text{fa}}^* \\ H_0, & \text{otherwise} \end{cases} \tag{29}$$

B. Naive Bayes (NB)

The Naive Bayes (NB) classifier makes a simplifying assumption regarding the estimated energy level on each SU, by considering that y_i, for $i = 1$ to N, are mutually independent. Thus one can write their joint conditional probability density function (pdf) as the multiplication of each marginal pdf:

$$f(\mathbf{y}|H_j) = \prod_{i=1}^{N} f(y_i|H_j) \text{ with } j \in \{0,1\} \tag{30}$$

Furthermore, by considering that $f(y_i|H_j)$ is Gaussian, $f(\mathbf{y}|H_j)$ becomes a multivariate Gaussian distribution with a diagonal covariance matrix. Indeed, as the number of samples K increases, $z_i(k)$ under hypothesis H_0 can be well approximated by a Gaussian distribution with mean K and variance $2K$. Likewise, $z_i(k)$ under hypothesis H_1 can be well approximated by a Gaussian distribution with mean $K(1+\gamma_i)$ and variance $2K(1+\gamma_i)^2$.

Considering a training set with M samples of $\mathbf{y}$ and $\mathbf{d}$, the Naive Bayes model can estimate the probability density functions parameters of each hypothesis by applying the maximum likelihood principle:

$$\begin{aligned} \hat{\mu}_i^{\{H_{1|0}\}} &= \frac{1}{M^{\{H_{1|0}\}}} \sum_{m=1}^{M^{\{H_{1|0}\}}} y_i[m]^{\{H_{1|0}\}} \\ \hat{\sigma_i^2}^{\{H_{1|0}\}} &= \frac{1}{M^{\{H_{1|0}\}}-1} \sum_{m=1}^{M^{\{H_{1|0}\}}} \left(y_i[m]^{\{H_{1|0}\}} - \hat{\mu}_i^{\{H_{1|0}\}}\right)^2 \end{aligned} \tag{31}$$

where the superscript $\{H_{1|0}\}$ refers to the section of the training set which comprises hypothesis H_1 or H_0.

After the *training phase*, the Naive Bayes classifier can estimate the channel occupancy *a posteriori* probability given the detected energy $\mathbf{y}$ through Bayes Theorem:

$$P(H_1|\mathbf{y}) = \frac{f(\mathbf{y}|H_1)P(H_1)}{f(\mathbf{y}|H_1)P(H_1) + f(\mathbf{y}|H_0)P(H_0)} \tag{32}$$

where $P(H_1)$ and $P(H_0)$ are the *a priori* probabilities of each hypothesis, estimated as follows.

$$P(H_1) = \frac{M^{\{H_1\}}}{M} \quad (33) \qquad P(H_0) = \frac{M^{\{H_0\}}}{M} \quad (34)$$

where $M^{\{H_i\}}$, $i = 0, 1$ is the number of the ith hypothesis occurrences.

Finally, the Naive Bayes channel status is inferred based on the evaluation:

$$\hat{S}_{\text{NB}} = \begin{cases} H_1, & \text{if } P(H_1|\mathbf{y}) \geq 1 - P_{\text{fa}}^* \\ H_0, & \text{otherwise} \end{cases} \tag{35}$$

C. Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a classification technique aiming at finding a linearly separable hyperplane with a maximum margin between the classes by applying a *kernel* function $\kappa(\mathbf{x}, \mathbf{x}')$ to the input vector in order to increase its dimension from *input space* to *feature space*. It is worth noting that for the SVM the desired channel status needs to be $d_m \in \{-1; 1\}$, representing hypothesis H_0 and H_1, respectively.

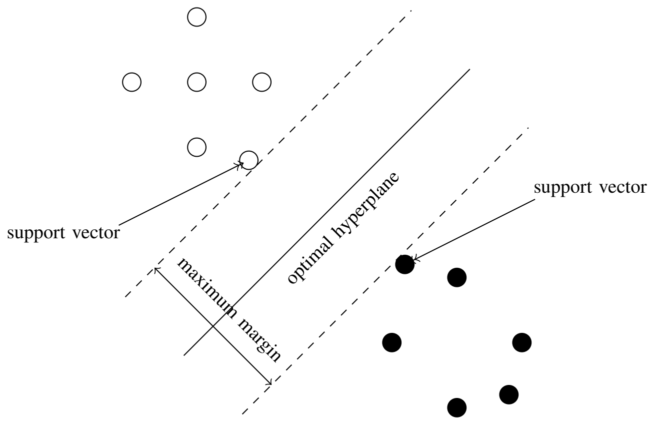


Figure 4. Illustration of a support vector machine decision plane. The optimal hyperplane separates the two classes (filled circles and empty circles) at half distance of the maximum margin defined by the support vectors.

Therefore, given the following linear model:

$$h(\mathbf{y}) = \mathbf{w}^T \phi(\mathbf{y}) + b \tag{36}$$

where $\phi(\cdot)$ is a feature-space transformation function, our goal is to find $\mathbf{w}$ and b such that $h(\mathbf{y}_m) > 0$ if $d_m = 1$ and $h(\mathbf{y}_m) < 0$ if $d_m = -1$. Moreover, one can apply (36) to maximize the decision margin of the hyperplane $h(\mathbf{y}) = 0$. By taking into account that the distance of any point $\mathbf{y}$ to the decision surface is given by:

$$\frac{d_m\left(\mathbf{w}^T \phi(\mathbf{y}_m) + b\right)}{\|\mathbf{w}\|} \tag{37}$$

we choose to define that for the closest point to the decision surface, $d_m(\mathbf{w}^T \phi(\mathbf{y}_m) + b) = 1$. Thus, the following optimization problem can be formulated [22]:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}}\ \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t.}\quad d_m(\mathbf{w}^T \phi(\mathbf{y}_m) + b) \geq 1, \qquad m = 1, \ldots, M \end{aligned} \tag{38}$$

where the constraint in (38) guarantees that every sample is correctly classified. The points $\mathbf{y}_m$ for which this constraint is met with the equality sign are called *support vectors*. The support vectors completely characterize the SVM model, while the rest of the training samples are entirely irrelevant.

Notwithstanding, Eq. (38) assumes that the classes can be perfectly separated in feature space, which is not always true in our spectrum sensing context. To address the case of overlapping classes, we can modify the constraint by introducing the slack variable δ_m and an additional *overlap budget* ξ. Therefore, (38) can be rewritten as [20] [12]:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}}\ \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t.}\quad (\text{c.1})\ d_m(\mathbf{w}^T \phi(\mathbf{y_m}) + b) \geq 1 - \delta_m,\ m = 1, \ldots, M \\ &\qquad\ (\text{c.2})\ \delta_m \geq 0,\ m = 1, \ldots, M \\ &\qquad\ (\text{c.3})\ \sum_{m=1}^{M} \delta_m \leq \xi \end{aligned} \tag{39}$$

where ξ is a constant used to control the trade-off between minimizing training errors and controlling the model complexity (which can be used as a heuristic to avoid overfitting).

The problem in Eq. (39) is quadratic with linear constraints, with the following Lagrange primal function [22]:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + \xi \sum_{m=1}^{M} \delta_m \\ - \sum_{m=1}^{M} \alpha_m \left[d_m h(\mathbf{y}_m) - 1 + \delta_m\right] - \sum_{m=1}^{M} \mu_m \delta_m \end{aligned} \tag{40}$$

where α_m and μ_m are Lagrange multipliers.

By setting the derivatives to zero, we have:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{m=1}^{M} \alpha_m d_m \phi(\mathbf{y}_m) \tag{41}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{m=1}^{M} \alpha_m d_m = 0 \tag{42}$$

$$\frac{\partial \mathcal{L}}{\partial \delta_m} = 0 \rightarrow \alpha_m = \xi - \mu_m \tag{43}$$

From (40)-(43) we have the following set of KKT conditions:

$$\alpha_m \geq 0 \tag{44a}$$

$$d_m h(\mathbf{y}_m) - 1 + \delta_m \geq 0 \tag{44b}$$

$$\alpha_m \left(d_m h(\mathbf{y}_m) - 1 + \delta_m\right) = 0 \tag{44c}$$

$$\mu_m \geq 0 \tag{44d}$$

$$\delta_m \geq 0 \tag{44e}$$

$$\mu_m \delta_m = 0 \tag{44f}$$

By substituting (41)–(44) into (40) we can obtain the dual problem w.r.t. the support vector:

$$\tilde{\mathcal{L}}(\boldsymbol{\alpha}) = \sum_{m=1}^{M} \alpha_m - \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{M} \alpha_m \alpha_n d_m d_n \kappa(\mathbf{y}_m, \mathbf{y}_n) \tag{45}$$

where $\kappa(\mathbf{y}_m, \mathbf{y}_n) = \phi(\mathbf{y}_m)^T \phi(\mathbf{y}_n)$ is a kernel function such as the linear $\mathbf{y}_m^T \mathbf{y}_n$. A kernel function, in turn, can be formally defined as [23]: *a function that computes the inner product of the images produced in the feature space under the embedding ϕ of two data points in the input space.*

Finally, one can formulate the following optimization problem [12] [22] [24]:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\text{maximize}} && \tilde{\mathcal{L}}(\boldsymbol{\alpha}) \\ &\text{s.t.} \quad \text{(c.1)} && 0 \leq \alpha_m \leq \xi \\ &\quad\quad\ \ \text{(c.2)} && \sum_{m=1}^{M} \alpha_m d_m = 0 \end{aligned} \tag{46}$$

which can be solved using standard quadratic programming (QP) techniques.

By defining $\boldsymbol{\alpha}^\star$ as the solution to the dual problem (46) and $b^\star$ as the solution of the primal problem (39), as well as modifying (36) to be expressed in terms of $\boldsymbol{\alpha}^\star$, we can obtain the output of the SVM to an unseen example according to:

$$h(\mathbf{y}) = \sum_{m=1}^{M} \alpha_m^\star d_m \kappa(\mathbf{y}, \mathbf{y}_m) + b^\star \tag{47}$$

It is worth noting that in order to predict the output for an unseen example, the SVM retains only the support vectors of the training dataset, i.e., where α_m is nonzero.

Once we obtain the SVM output $h(\mathbf{y})$, we can decide the channel status by first converting the output to the estimated *a posteriori* probability $\hat{P}(H_1|\mathbf{y})$ as [25]:

$$\hat{P}(h(\mathbf{y})) = \frac{1}{1 + e^{(Ah(\mathbf{y})+B)}} \tag{48}$$

where the parameters A and B can be found by minimizing the negative log likelihood function (LLF) of the training data:

$$\text{minimize} \quad -\sum_{m} t_m \log(p_m) + (1 - t_m) \log(1 - p_m) \tag{49}$$

where $p_m = \dfrac{1}{1 + e^{(Ah(\mathbf{y}_m)+B)}}$, and $t_m = \dfrac{d_m + 1}{2}$.

Finally, the channel status inference using SVM approach results:

$$\hat{S}_{\text{SVM}} = \begin{cases} H_1, & \text{if } \hat{P}(h(\mathbf{y})) \geq 1 - P_{\text{fa}}^* \\ H_0, & \text{otherwise} \end{cases} \tag{50}$$

V. Numerical Results

In order to assess the performance of the models mentioned above on channel status inference, we ran Monte-Carlo simulations (MCS) with 5×10^4 realizations, considering a scenario with one PU and 3 SUs depicted at Fig. 5, as well as the parameters on Table I under AWGN and Rayleigh flat fading channels.

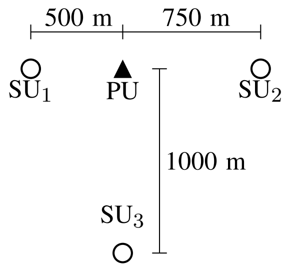


Figure 5. Cognitive radio network scenario for evaluation purpose.

We compared the machine learning methods with traditional analytical methods AND, OR and MRC. For the SVM, we have considered both the linear and Gaussian kernel functions. As for the MLP, we considered a network of one hidden layer with size equal to the number of inputs and only one output.

Table I. Adopted system parameters in the Monte Carlo simulations.

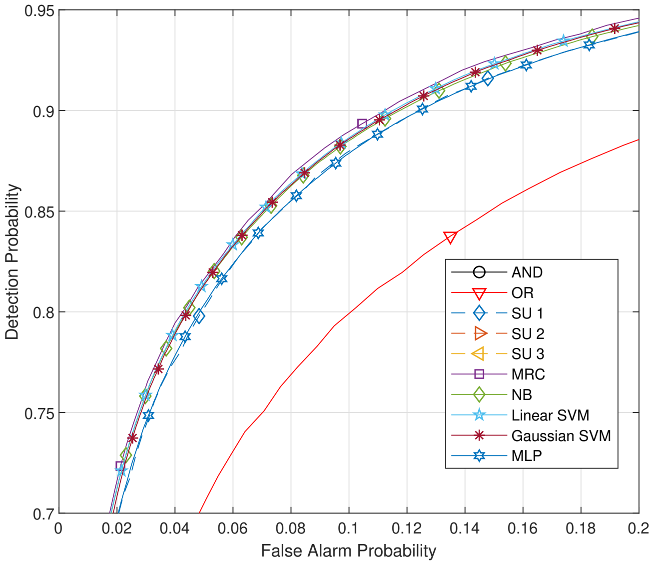
Parameters	Value
Bandwidth	$w = 5$ MHz
Sampling frequency	$f_s = 10$ MHz
Noise PSD	$\eta_0 = -152$ dBm/Hz
PU active probability	$P(H_1) = 0.5$
PU transmission power	$\sigma_s^2 = 0.1$ mW
SU$_1$ $\rightarrow$ PU distance	500m
SU$_2$ $\rightarrow$ PU distance	750m
SU$_3$ $\rightarrow$ PU distance	1000m
Sensing time-interval	$\tau = 5\mu s$
Number of samples	$K = 2w\tau = 50$
Training dataset size	$\kappa \in [50; 100; 250; 500; 1000]$

A. SS Performance under AWGN Channels

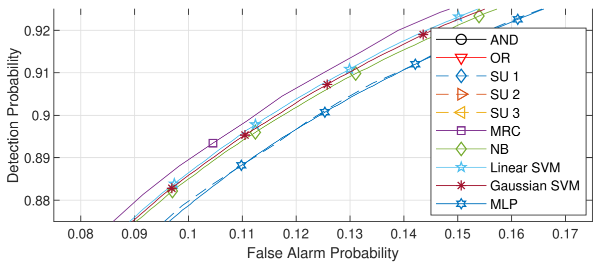
For each model, we evaluated the receiver operating characteristic (ROC) curve, depicted in Fig. 6a and 6b. By visual inspection, we can notice the upper bound on performance defined by the MRC technique, followed closely by the SVM with a linear kernel.

Alongside the cooperative techniques, we have the plot of the ROC curves obtained by individual energy detection on each SU. Due to the PU distance differences, the average SNR level obtained were $\bar{\gamma}_1 \approx -2dB$, $\bar{\gamma}_2 \approx -9dB$ and $\bar{\gamma}_3 \approx -14dB$. This difference becomes apparent on the channel detection performance displayed by each SU.

In order to better evaluate the results of each model, we also obtained the area under the curve (AuC) metric



(a) ROC curve



(b) Zoom into the ROC $0.9P_{\mathrm{d}}$ / $0.1P_{\mathrm{fa}}$ interest region.

Figure 6. Receiver operating characteristic (ROC) curves for the different techniques under AWGN channel.

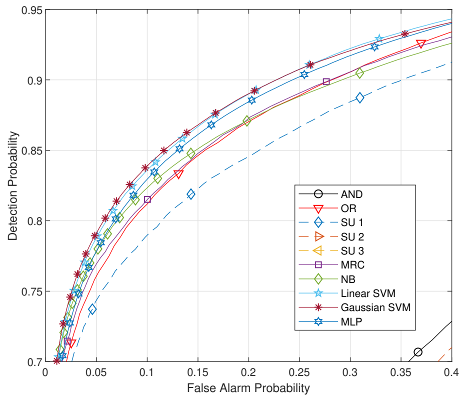
in Table II. Hence, from Fig. 6 and Tab. II we notice the machine learning methods perform much better than the more straightforward AND or OR techniques, with results very close to MRC centralized SS technique.

Table II. AREA UNDER THE CURVE (AUC) RESULTS FOR AWGN AND RAYLEIGH CHANNELS

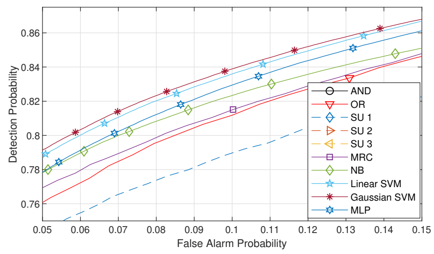
TECHNIQUE	AUC AWGN	AUC Rayl.
AND	0.7186	0.7370
OR	0.9302	0.9256
MRC	**0.9616**	0.9240
NB	0.9594	0.9244
SVM-Linear	**0.9613**	**0.9360**
SVM-Gaussian	**0.9604**	**0.9327**
MLP	**0.9609**	**0.9359**

B. *SS Performance under Rayleigh Fading Channels*

On the other hand, when considering a flat Rayleigh fading channel, the ML techniques were able to attain better performance than MRC, as it can be seen in Fig. 7 and Table II, because MRC considers the average SNR level over each secondary user, which, under a flat Rayleigh fading channel, varies for each sensing period. Indeed, from Tab. II, one can notice that the MLP and linear-SVM have achieved the highest AuC values, similarly from the Gaussian channel results.



(a) ROC curve



(b) Zoom into the ROC $0.9P_{\mathrm{d}}$ / $0.1P_{\mathrm{fa}}$ interest region.

Figure 7. Receiver operating characteristic (ROC) curves for the different techniques under Rayleigh fading channel.

In order to assess the effects of the amount of training samples on the final AUC metric under Rayleigh channel, we variated training set size from 50 to 1000 samples. Fig. 8 shows the resulting for the AUC variation. Clearly, all analyzed ML techniques benefit from the increase of the training set size, with the sharpest difference between 50 and 500 samples. It is apparent that the suitable performance-complexity tradeoff occurs for $\kappa = 100$ training samples for the SVM Linear and Gaussian and $\kappa = 250$ training samples for the MLP and NB machine learning techniques.

C. *Complexity*

In order to compare the computational burden complexity of each technique, we obtained the time spent during the training and inference phase for a dataset of $\kappa = 500$ training samples, averaged over 20 rounds. Considering that all models output *a posteriori* probability of channel occupancy, while the inference phase has significantly less computation time compared to the training phase. Table III summarizes the time spent for each SS technique in both training and inference phases of the analyzed CRN scenario of Fig. 5.

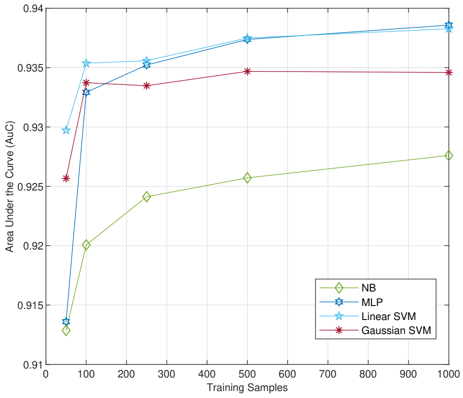


Figure 8. Variation of Area Under the Curve results for different κ training samples for the SS-CRNs operating under Rayleigh channels.

Table III. AVERAGE TIME SPENT DURING TRAINING PHASE

SS TECHNIQUE	TRAINING [sec]	INFERENCE [sec]
NB	0.420	0.141
SVM-Linear	1.060	0.035
SVM-Gaussian	0.327	0.082
MLP	**0.230**	0.024
MRC	-	**0.006**

As a drawback of increasing the size of the training set, Fig. 9 shows the increase in time spent during the training phase for the techniques. It is worth noting that only the naive Bayes model remains with almost constant training time for increased training sets.

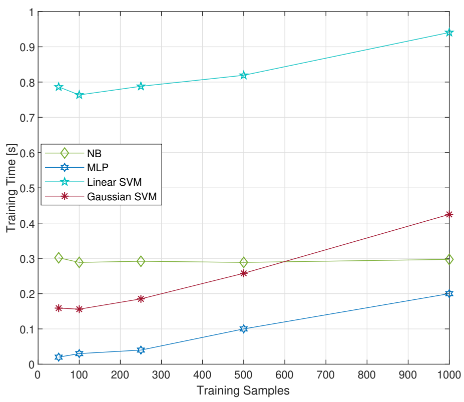


Figure 9. Time spent during training phase for different sizes of training sets.

The computational metrics were evaluated on a consumer laptop with Intel i7 5500U @2.4 GHz and 16GB DDR3 @1600 MHz RAM. Based solely on the results from Table III and the AuC metrics from Tab. II, the MLP achieves the best performance-computational complexity trade-off operating in both AWGN or Rayleigh channels when compared to the others machine learning models.

Nonetheless, it is worth noting that besides time spent training, the machine learning techniques have the requirement of cooperation from the PU to provide the fusion center with the channel status vector, which must be taken into account a system complexity of implementing such models.

On the other hand, while the MRC technique provides almost instantaneous channel status inference based on energy samples, it requires the SUs transmit to the fusion center in a completely centralized way their estimated SNR, which can also pose a further challenge for implementation while reduce the overall spectral efficiency of the CRN.

VI. DISCUSSION AND FINAL REMARKS

In this paper, we compared the use of machine learning models, namely the support vector machine with linear and Gaussian kernel functions, a feed-forward neural network with one hidden layer and Naive Bayes method with those well-established analytical models in the context of spectrum sensing problem for cognitive radios.

Numerical results demonstrated that the studied models are of proved suitability for the task of channel status inference from energy samples obtained from multiple secondary users. Based on the *receiver operating characteristic* curve and *area under the curve* metrics, we are led to conclude that all machine learning models performed closely to the optimum maximum ratio combining analytical technique under AWGN channels.

Interestingly enough, under Rayleigh flat fading channels, all machine learning techniques outperformed the MRC due to varying SNR levels on secondary users over each sensing period. It becomes clear that MRC needs to perform channel response estimation in order to be able to perform well on fading channels, which could be a challenge under cognitive radio networks.

By using standard profiling tools, we were able to obtain computational performance metrics for each machine learning model evaluated during the training and inference phases. For small CRNs context, the results demonstrated an advantage of multilayer perceptron technique followed by the Gaussian support vector machine, being the fastest model to train and infer the channel status, achieving great AuC performance on channel status inference.

One interesting aspect that should also be considered in a future work is how often such machine learning models need to be re-trained under non-stationary channels (such as imposed by SU and/or PU mobility).

REFERENCES

[1] Y. Chen and H.-S. Oh, “A survey of measurement-based spectrum occupancy modeling for cognitive radios,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 848–859, October 2014.

[2] Z. Sun, G. J. Bradford, and J. N. Laneman, "Sequence detection algorithms for phy-layer sensing in dynamic spectrum access networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 97–109, September 2010.

[3] FCC, "Spectrum policy task force," *ET Docket No. 02-155*, November 2002.

[4] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, August 1999.

[5] F. hsun Tseng, L. der Chou, H. chieh Chao, and J. Wang, "Ultra-dense small cell planning using cognitive radio network toward 5g," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 76–83, December 2015.

[6] M. Jia, X. Gu, Q. Guo, W. Xiang, and N. Zhang, "Broadband hybrid satellite-terrestrial communication systems based on cognitive radio toward 5g," *IEEE Wireless Communications*, vol. 23, no. 6, pp. 96–106, December 2016.

[7] S. H. Chae, C. Jeong, and K. Lee, "Cooperative communication for cognitive satellite networks," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5140–5154, June 2018.

[8] B. Li, Z. Fei, Z. Chu, F. Zhou, K.-K. Wong, and P. Xiao, "Robust chance-constrained secure transmission for cognitive satellite–terrestrial networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4208–4219, January 2018.

[9] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. The MIT Press, 2014.

[10] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, Jun 2018. [Online]. Available: https://doi.org/10.1186/s13174-018-0087-2

[11] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 126, pp. 1–32, 2019. [Online]. Available: https://doi.org/10.3390/s19010126

[12] 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, October 2013.

[13] F. Azmat, Y. Chen, and N. Stocks, "Analysis of spectrum occupancy using machine learning algorithms," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 6853–6860, October 2015.

[14] Z. Li, W. Wu, X. Liu, and P. Qi, "Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks," *IET Communications*, vol. 12, no. 19, pp. 2485–2492, November 2018.

[15] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1136–1159, October 2012.

[16] H. A. Shah and I. Koo, "Reliable machine learning based spectrum sensing in cognitive radio networks," *Wireless Communications and Mobile Computing*, no. ID 5906097, p. 17, 2018. [Online]. Available: https://doi.org/10.1155/2018/5906097

[17] Z. Li, W. Wu, X. Liu, and P. Qi, "Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks," *IET Communications*, vol. 12, no. 19, pp. 2485–2492, 2018.

[18] Y. Lu, P. Zhu, D. Wang, and M. Fattouche, "Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks," in *2016 IEEE Wireless Communications and Networking Conference*, April 2016, pp. 1–6.

[19] R. Umar, A. U. H. Sheikh, and M. Deriche, "Unveiling the hidden assumptions of energy detector based spectrum sensing for cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 713–728, September 2013.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[21] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, pp. 386–408, 1958.

[22] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer New York Inc., 2001.

[23] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[24] S. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson Education, 2009.

[25] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

References

ALPAYDIN, E. *Introduction to Machine Learning*. 3rd. ed. [S.l.]: The MIT Press, 2014.

ARJOUNE, Y.; KAABOUCH, N. A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions. *Sensors*, v. 19, n. 126, p. 1–32, 2019.

AZMAT, F.; CHEN, Y.; STOCKS, N. Analysis of spectrum occupancy using machine learning algorithms. *IEEE Transactions on Vehicular Technology*, v. 65, n. 9, p. 6853–6860, October 2015.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006. ISBN 0387310738.

BKASSINY, M.; LI, Y.; JAYAWEERA, S. K. A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys and Tutorials*, v. 15, n. 3, p. 1136–1159, October 2012.

BOUTABA, R.; SALAHUDDIN, M. A.; LIMAM, N.; AYOUBI, S.; SHAHRIAR, N.; ESTRADA-SOLANO, F.; CAICEDO, O. M. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, v. 9, n. 1, p. 16, Jun 2018. ISSN 1869-0238.

CHAE, S. H.; JEONG, C.; LEE, K. Cooperative communication for cognitive satellite networks. *IEEE Transactions on Communications*, v. 66, n. 11, p. 5140–5154, June 2018.

CHEN, Y.; OH, H.-S. A survey of measurement-based spectrum occupancy modeling for cognitive radios. *IEEE Communications Surveys & Tutorials*, v. 18, n. 1, p. 848–859, October 2014.

FCC. Spectrum policy task force. *ET Docket No. 02-155*, November 2002.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. 2nd. ed. [S.l.]: Springer New York Inc., 2001.

HAYKIN, S. S. *Neural Networks and Learning Machines*. 3rd. ed. [S.l.]: Pearson Education, 2009.

JIA, M.; GU, X.; GUO, Q.; XIANG, W.; ZHANG, N. Broadband hybrid satellite-terrestrial communication systems based on cognitive radio toward 5g. *IEEE Wireless Communications*, v. 23, n. 6, p. 96–106, December 2016.

LI, B.; FEI, Z.; CHU, Z.; ZHOU, F.; WONG, K.-K.; XIAO, P. Robust chance-constrained secure transmission for cognitive satellite–terrestrial networks. *IEEE Transactions on Vehicular Technology*, v. 67, n. 5, p. 4208–4219, January 2018.

LI, Z.; WU, W.; LIU, X.; QI, P. Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks. *IET Communications*, v. 12, n. 19, p. 2485–2492, November 2018.

LI, Z.; YU, F. R.; HUANG, M. A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios. *IEEE Transactions on Vehicular Technology*, v. 59, n. 1, p. 383–393, September 2009.

LU, Y.; ZHU, P.; WANG, D.; FATTOUCHE, M. Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks. In: *2016 IEEE Wireless Communications and Networking Conference*. [S.l.: s.n.], 2016. p. 1–6.

MA, J.; ZHAO, G.; LI, Y. Soft combination and detection for cooperative spectrum sensing in cognitive radio networks. *IEEE Transactions on Wireless Communications*, v. 7, n. 11, p. 4502–4507, December 2008.

MITOLA, J.; MAGUIRE, G. Q. Cognitive radio: making software radios more personal. *IEEE Personal Communications*, v. 6, n. 4, p. 13–18, August 1999.

PLATT, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers*. [S.l.]: MIT Press, 1999. p. 61–74.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 386–408, 1958.

SHAH, H. A.; KOO, I. Reliable machine learning based spectrum sensing in cognitive radio networks. *Wireless Communications and Mobile Computing*, p. 17, 2018.

SHAWE-TAYLOR, J.; CRISTIANINI, N. *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004. ISBN 0521813972.

SUN, Z.; BRADFORD, G. J.; LANEMAN, J. N. Sequence detection algorithms for phy-layer sensing in dynamic spectrum access networks. *IEEE Journal of Selected Topics in Signal Processing*, v. 5, n. 1, p. 97–109, September 2010.

THILINA, K. M.; CHOI, K. W.; SAQUIB, N.; HOSSAIN, E. Machine learning techniques for cooperative spectrum sensing in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, v. 31, n. 11, p. 2209–2221, October 2013.

TSENG, F. hsun; CHOU, L. der; CHAO, H. chieh; WANG, J. Ultra-dense small cell planning using cognitive radio network toward 5g. *IEEE Wireless Communications*, v. 22, n. 6, p. 76–83, December 2015.

UMAR, R.; SHEIKH, A. U. H.; DERICHE, M. Unveiling the hidden assumptions of energy detector based spectrum sensing for cognitive radios. *IEEE Communications Surveys & Tutorials*, v. 16, n. 2, p. 713–728, September 2013.