# On Spectrum Sensing, a Machine Learning Method for Cognitive Radio Systems

Youness Arjoune and Naima Kaabouch
School of Electrical Engineering and Computer Science
University of North Dakota
Grand Forks, North Dakota, USA

**Abstract—Spectrum sensing plays an important role in enabling cognitive radio technology for the up-and-coming generation of wireless communication systems. Over the last decade, several sensing methods have been proposed, including energy detection, cyclostationary feature, and matched filter. However, these techniques present several limitations. Energy detection performs poorly under low signal-to-noise ratio, cyclostationary features are complex, and matched filter requires some prior knowledge about the primary user signal. In addition, all of these techniques require setting a threshold which needs the prior knowledge of the noise distribution. Thus, the reliability of spectrum sensing is still an open issue in wireless communication research. In this paper, we propose a spectrum sensing method based on a machine learning theory for cognitive radio networks. The spectrum sensing problem is rigorously modeled and out of which a large-scale comprehensive dataset is built. This dataset is then used to train, validate, and test several machine learning techniques, including random forest, support vector machine with different kernels, decision tree, Naïve Bayes, K-nearest neighbors, and logistic regression. The models were extensively tested and evaluated using metrics such as the probabilities of detection, false alarm, and miss-detection as well as the accuracy of the classification. The simulation results show that the random forest model outperforms all the other machine learning methods.**

***Keywords—cognitive radio, machine learning, spectrum sensing, probability of detection, probability of false alarm***

## I. INTRODUCTION

Radio spectrum is a limited resource that necessities to be well managed. To a large extent, the Federal Communications Commission (FCC) policies regarding the radio spectrum are based on static management in which the radio spectrum is divided up and doled out for use in the form of licenses. This static management in the wake of the dramatic growth of wireless communication has created a shortage in the available spectrum. Cognitive radio has the ability to address this issue through dynamic management of the radio spectrum to ensure fair access among all users [1-6]. A cognitive radio is able to sense its surroundings and decide on the presence of signals and then adapt its parameters to maximize it transmission throughput without causing any interference to licensed users' signals. The FCC commission is closely following the development of these new technologies, and opened up the TV White Space for everyone making this portion of the spectrum the first real spectrum using the cognitive radio technology.

One of the main key functions of a cognitive radio system is spectrum sensing [7-9]. This function allows secondary users, unlicensed users, to sense the radio spectrum and identify free channels. Since its introduction, researchers have been working on developing novel techniques for spectrum sensing, and as a result, several methods have been proposed. Examples of these techniques include energy detection [10], cyclostationary detection [10, 11], and matched filter detection [12]. However, these methods present several limitations such as the inefficiency of energy detection under low values of signal-to-noise ratio, the complexity of cyclostationary detection, and the impracticality of matched filter detection [10]. In addition, all these techniques require setting an appropriate threshold which highly impacts their detection performance.

To address the limitations of above techniques, several papers have been hustling to investigate the use machine learning classifiers to detect the state of the primary user, especially in the context of cooperative spectrum sensing [14-19]. These approaches can be classified into two categories. Techniques in the first category use unsupervised and supervised machine learning techniques. For instance, the authors of [14-15, 18] proposed a two-step machine learning model for spectrum sensing. In the first step, the K-means algorithm is used to identify the state of the primary user present. In the second step, support vector machine or a similar type of classifier is used to collect the new input data into one of the classes specified by the K-means process in the first step. Techniques in the second category assume that the sensing classes are known and are based on a single step in which supervised machine learning techniques are used to train models. For instance, the authors of [17, 19] used a single step method in which supervised machine-learning classifiers—K-nearest neighbor, support vector machine, Naïve Bayes, and decision tree—were applied. In several other works [14, 17, 19], the authors used multiple classifiers to train their models then selected the best classifier for their models.

Most of these previous machine learning models did not perform feature selection, which is necessary in the context of machine learning theory as this process allows the selection of the most relevant features that enable the classifiers to have high accuracies. Feature selection is not always a straightforward operation.

The theory of machine learning itself has made significant advances and has been applied to solve many complex problems. In this paper we investigate the efficiency of random forest as a technique for spectrum sensing. This efficiency is compared to several other machine learning techniques, including support vector machine with different kernels, decision tree, k-nearest neighbors, Naïve Bayes, and logistic regression. The main contribution of this paper can be summarized as follows:


978-1-7281-0927-5/19/$31.00 ©2019 IEEE

- A comprehensive, up-to-date review of the key research works on spectrum sensing
- Modeling rigorously the problem of spectrum sensing in cognitive radio network
- Building a large-scale comprehensive dataset
- Training, validating, and testing several machine learning techniques including random forest, support vector machine with different kernels, decision tree, Naïve Bayes, K-nearest neighbors, and logistic regression
- Performance comparison of these machine learning techniques using several evaluation metrics including the probabilities of detection, the false alarm, and miss-detection as well as the accuracy.

The remaining sections of this paper are organized as follows: Section 2 provides the mathematical model of spectrum sensing and describes convolutional neural networks. Section 3 presents the simulation setup, evaluation metrics, and examples of the obtained results. Sections 4 provides conclusions and future research directions.

## II. METHODOLOGY

This section describes the mathematical model for spectrum sensing as well as the methodology for spectrum sensing using machine learning techniques. Spectrum sensing's mathematical model is presented first, then the mathematical models of the machine learning techniques.

### *A. Spectrum Sensing*

We first consider a cognitive radio network in which there are several secondary users and a single primary user randomly distributed in a given area. The primary user can randomly use its licensed frequency channel and secondary users are continuously monitoring the state of this channel by performing spectrum sensing and accessing the channel when it is idle. Furthermore, the primary user is assumed to be transmitting a signal with a fixed power P.

Spectrum sensing can be modeled using hypothesis testing. The first hypothesis corresponds to the absence of the primary user signal, denoted $H_a$, while the second one corresponds to the presence of the primary user signal, denoted $H_p$ [20-21].

Let $x(t)$ denote the primary user signal and let $y(t)$ denote the received signal at the receiver of the secondary user. Thus, spectrum sensing can be expressed as:

$$\begin{cases} H_a: y(t) = n(t), \\ H_p: y(t) = x(t) + n(t), \end{cases} \tag{1}$$

where n(t) denotes additive white Gaussian noise.

To perform spectrum sensing, the energy statistic is used as a feature to train the classifiers and determine the state of these frequency channels. This process consists of two steps: the first uses k-means clustering, an unsupervised machine-learning technique to define these two classes. Then the elements of the dataset are labeled. This dataset is used to train several supervised machine learning classifiers in the second step, which must learn to classify each channel based on these features.

The energy statistic refers to the squared magnitude of the fast Fourier transform of the received signal averaged over the number of samples received. It is given by:

$$E_n = \frac{1}{N}\sum_{n=1}^{N}(R[n])^2, \tag{2}$$

where N is the total number of the received samples, and $R[n]$ is the $n^{th}$ element of the vector formed by the fast Fourier transform of the received samples.

### *B. Machine learning techniques*

K-means clustering: Among the cluster data techniques, K-means clustering is one of the more popular methods that has merit. This method uses the energy statistic to group the frequency channels into one of two classes: "channel free" and "channel busy". It begins by initializing of the centroid of the two clusters and then maps each channel to the class that minimizes the identification errors of each measurement and the centroids by solving optimization problems, which can be formulated as:

$$\min_{\{mc\}}\sum_{i}^{N}\|E_n - m_c\|_2^2, \tag{3}$$

where $E_n$ denotes the energy statistic of the received samples and $m_c$ denotes the cluster centroid.

Once the channels are mapped into one of the two clusters, the corresponding classes are identified and each channel is labeled either as free or busy. Then, different supervised machine learning methods are used for classification as with support vector machine.

Support vector machine: a method that creates a hyperplane to separate data of two classes. The choice of the kernel determines the separation boundary between the classes. Different kernels such as linear kernel, radial basis function, quadratic, and cubic kernels can be used with support vector machine.

The linear kernel is defined as:

$$K(E) = \mathrm{w}^T E + b, \tag{4}$$

where $E$ is the vector formed by energy statistic of the received samples corresponding to all the frequency channels. w denotes the vector of weights and $(.)^T$ denotes the transpose operator, and b is the bias vector.

The linear support vector machine is formulated as solving an optimization problem over the weights $w$ such as:

$$\min_{w\in R^d}\|w\|^2 + C\sum_i^N \max(0, 1 - y_i K(E_i)), \tag{5}$$

where $y_i$ is the $i^{th}$ element of the output, C is a regularization constant, and $E_i$ is the energy statistic of the $i^{th}$ element in the dataset.

The quadratic and cubic kernels are polynomial kernels with degrees of 2 and 3, respectively. Polynomials kernels are defined as:

$$K(E, y) = (E.y + 1)^d, \quad (6)$$

where $E$ is the vector of energy statistic of the received samples of all frequency channels undergoing sensing, $y$ is the output vector whose elements take two values 0 and 1, and d is the degree of the polynomial.

Radial basis function kernel is defined as:

$$K(E, y) = \exp(-\gamma \|E - y\|^2), \quad (7)$$

where $E$ is the vector of energy statistic, y is the output vector, and $\gamma$ is a constant.

Logistic regression: A logistic regression technique analyzes datasets based on one or more independent variables that produce an outcome. The outcome contains only two classes denoted by data code as 1 or 0. The logistic regression hypothesis can be defined as:

$$h_\theta(E) = g(\theta^T E), \quad (8)$$

where $E$ is the vector of energy statistic, $\theta$ is the vector of weights, $(\cdot)^T$ denotes the transpose operator, and $g$ is the sigmoid function, which is given by:

$$g(\theta^T E) = \frac{1}{1+e^{-\theta^T E}}. \quad (9)$$

The cost function of logistic regression is given by:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)} \log\left(h_\theta\left(E^{(i)}\right)\right) - (1 - y^{(i)})\log(1 - h_\theta\left(E^{(i)}\right))], \quad (10)$$

where $m$ is the training size, $h_\theta(\cdot)$ is the hypothesis function given by equation (9), $E^{(i)}$ the $i^{th}$ element of the vector of the energy statistic, and $y^{(i)}$ is the $i^{th}$ elements of the training output whose possible values are either 0 or 1.

Observed class labels and energy statistic features are used to calculate the unknown weights $\theta$ for logistic regression by minimizing the previous cost function. Gradient descent is one of the most popular techniques used to minimize this cost function. The optimization function is thus written as:

$$\min_\theta J(\theta) \quad (11)$$

where J(.) is the cost function and $\theta$ is the vector of weights.

**Naïve Bayes:** Naïve Bayes is a simple classification technique that uses Bayes' theorem with an assumption of independence of the input feature. In short, a naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. This classifier is known by its power to deal with large dataset and can even outperform highly sophisticated classification methods. Bayes' theorem provides a way of calculating the posterior probability distribution of the output vector y or the target class given the input vector of features, which is in our case the energy statistic.

$$P(y|E) = \frac{p(E|\, y)p(y)}{p(E)}, \quad (12)$$

where $P(y|\mathrm{E})$ is the posterior probability of the output y given the input vector of energy statistic feature, $p(E)$ is the prior probability of the input vector of energy statistic feature E, $p(y)$ is the prior probability of output vector $y$, and $p(E|\, y)$ is the likelihood function which is the probability of the input vector of the energy statistic feature.

Decision Tree: a hierarchical classifier technique. It is a simple and widely used classification method that uses a tree-like graph of decisions and possible outputs. The elements of the dataset are classified by sorting them as trees based on their features. Decision Tree does not require domain knowledge in which training data are collected and parameter setting is not required. Decision tree has three different types of nodes: root node, internal node, and leaf node. The decision node is the feature of the dataset to be classified and the branches are the values that the node can predict. In this paper, we build the decision tree using Iterative Dichotomiser 3 (ID3) which uses the entropy function and information gain as metrics to determine the attributes that best classify the training dataset. In order to define the information gain precisely, we start by defining the entropy that characterizes the purity of an arbitrary collection.

In this model, information gain is used to create attribute selection measures as follows:

$$H(s) = -\sum_{c\epsilon C} -p(c)\log_2 p(c), \quad (13)$$

where S is the current set formed by the vector of the energy statistic of the frequency channels for which the entropy is being calculated. C is the set of classes in S: C={channel free, channel busy} and $p$ is the proportion of the number of elements in class C to the number of elements in set S; when H(s) equals 0, set S is perfectly classified.

The information gain $IG(A)$is the measure of the difference in entropy from before to after the set S is split on an attribute A. This can be expressed as

$$IG(A, S) = H(S) - \sum_{t\in T} p(t)H(t), \quad (14)$$

where H(s) is the entropy of the set S, T is the set S by attribute A such that S is the union of the all splits: $S = U_{t\epsilon T}(t)$, p(t) is the proportion of elements in t to the number of elements in set S, and H(t) is the entropy of the subset T.

In ID3, information gain can be calculated for each remaining attribute. The attribute with the largest information gain is used to split the set S on this iteration

Random forest: A random forest is one of the most accurate classifiers. Random forest is based on the decision tree algorithm as it uses an ensemble of trees for the classification explained in the decision tree algorithm. To develop enriched ensembles, random vectors are constructed, which help to generate each tree from each random vector. Random forest uses the output of trees to solve the classification problem. Each tree votes to predict a decision and a majority determines the final random forest prediction. The following function is used to predict the output [38]:

$$Prediction = \frac{1}{B}\sum_{b=1}^{B} f_b(E), \quad (15)$$

where E is the vector energy statistic feature, $f$ is the prediction, $f_b$ is the trained classification or regression, B is the number of trees which is a free parameters.

**K-nearest neighbors:** K-Nearest Neighbor falls into the supervised machine learning family of algorithms. The KNN classifier is a non-parametric instance-based classification method where the algorithm does not make any assumptions about the underlying distribution of the dataset and classifies the input data based on the principle of similarities between the instances. The similarity between instances can be calculated as Euclidean distance between the instance of testing data and the nearest instance of training data [39]. This distance is given by:

$$d(E_i, E_j) = \sqrt{a\sum(E_{i,a} - E_{j,a})^2}, \quad (16)$$

where d is the distance between training and testing data, $x_i$ is the training dataset and $x_j$ is the testing dataset. $a$ is the data example.

Given a positive integer K, the energy statistic of a frequency channel, and a similarity metric d, the KNN classifier performs two steps: 1) It runs through the whole dataset computing the distance between $E_i$ and each training observation. The K points in the training data that are closest to $E_i$ form a set $\mathrm{A}$. K is usually odd to prevent tie situations. 2) It then estimates the conditional probability for each class, the fraction of points in A with that given class label. This probability can be expressed as:

$$P(y = j|E = E_i) = \frac{1}{K}\sum_{i\epsilon\,\mathrm{A}} I(y^{(i)} = j), \quad (17)$$

where $I(x)$ is an indicator function which evaluates to 1 when the argument $x$ is true and 0, otherwise. K is a positive integer, and j is the label of the output y, which is either 0 or 1.

## III. Results

To validate the proposed Model, a dataset of size 5000 was generated. To generate this dataset, we used a cognitive radio network model in which the primary user is operating under two scenarios: the first corresponds to the presence of the primary user signal in which the primary user is transmitting a signal while the second corresponds to the absence of the primary user signal. In both scenarios, the secondary users are performing spectrum sensing and are distant from the primary user with random distance in a geographic area of $1\,km^2$. These secondary users are experiencing different noise levels and the quality of the channels is time-varying. This dataset is then used to train machine learning techniques.

To evaluate the performance of the machine learning based spectrum sensing techniques, we used probabilities of detection, miss-detection, false alarm, and accuracy. The probability of detection refers to the likelihood of declaring that the primary user signal is present when it is actually present. This is calculated as the ratio between the number of times the primary user signal is declared present and the total number of time the primary user is present. It is given by:

$$Pd = \frac{\text{Number of times the PU is detected active}}{\text{Total number the PU is effectively active}} \quad (18)$$

The probability of miss-detection refers to the probability of detection that the primary user signal is absent when the primary user signal is active. It is measured as the ratio between the number of times the primary user is detected inactive while it is actually active. This can be expressed as:

$$Pm = \frac{\text{Number of times the PU is detected inactive}}{\text{Total number the PU is effectively active}} \quad (19)$$

The probability of false alarm refers to the likelihood that the technique detects the presence of the primary user while it is inactive. This is calculated as the ratio between the total number of the times the primary user is declared active while it is inactive. It is given by:

$$Pfa = \frac{\text{Numer of times PU signal is detected active}}{\text{Total number of times the PU is inactive}} \quad (20)$$

The accuracy of classification is another widely used metric in classification problems. It refers to the probability of detecting whether the PU is active when it is actually active and detecting that the PU is inactive when it is inactive. This can be formulated as follows:

$$A = \frac{\text{Number of times PU state is correctly detected}\times 100}{\text{Total number of trials}} \quad (21)$$

Several experiments were conducted. Examples of results are given in Figs .1 through 4. Fig. 1 shows the accuracy of the classification of channel states, free or busy, as a function of the number of folds. For fold numbers less than 10, the accuracy is around 99.2% which increases as the number of folds increases to reach its highest value, 99.64%, for a number of k-folds equal to 50. For K-folds higher than 50, this accuracy changes little and is approximately 99.65%.

Fig. 2 shows the accuracy of classification with random forest for both training and testing as a function of the training size. For training, this accuracy is maximum even for the lower training size and it remains same for the highest training size. However, for testing this accuracy is 98.2% for a training size equal to 100. As the training size increase, the testing accuracy of the classification increases also to reach a maximum value of 99.6% at a training size equal to 8000. The figure shows that the difference between the accuracy of the classification when tested on training data is very close to that tested on a new

dataset, meaning that the classifier does not suffer from high variance and low bias. In other words, the accuracy of random classifiers will remain above 99% if a new dataset is fed to the classifier.

The performance of random forest depends on the number of estimators used. To find the optimal number of estimators, we conducted several classifications with random forest using different numbers of estimators. Examples are given in Fig. 3 which show the accuracy of random forest versus the number of estimators for both training and testing dataset. Here, for a number of estimators equal 1, the accuracy of the classification is 99.8% when the model is tested on the training dataset while this accuracy is 99.5% when the model is tested on a new dataset. As the number of estimators increases, both accuracies increase to reach 99.95% and 99.6%, respectively, for 10 estimators. For estimators higher than 10, these accuracies remain nearly the same for training and testing: 99.97% and 99.63%, respectively. For all estimators, the difference between the training and testing accuracies of the classification is very small, meaning that the model will have the same performance if trained on a new dataset.

Fig. 4 shows the probability of detection as a function of the probability of false alarm for multiple algorithms. As one can see, the probability of detection increases as the probability of false alarm increases. Random forest, Naïve Bayes, and linear support vector machine outperform the rest of the algorithms in this measurement. Polynomial support vector machine with degree 2 have the lowest ROC curve, followed by support vector machine with radial basis function kernel.

Table 1 provides a performance comparison between random forest and the other classifiers such as support vector machine with different classifiers, logistic regression, K-nearest neighbors, and Naïve Bayes in terms of probabilities of detection, false alarm, and miss-detection as well as accuracy. From this table it can be seen that random forest has the highest probability of detection followed by linear support vector machine and Naïve Bayes. The effect of different kernels for

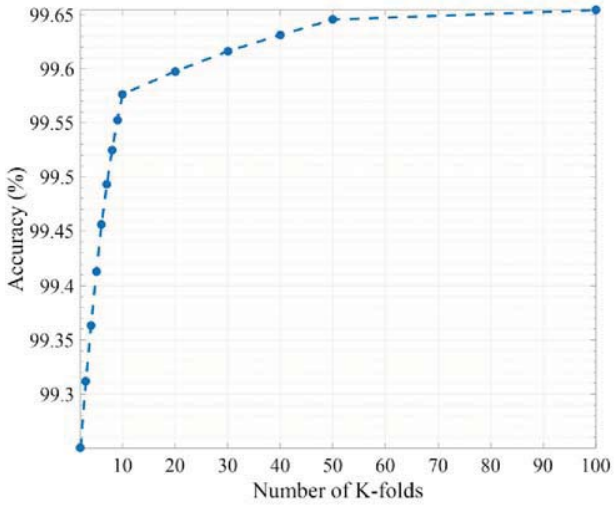


Fig. 1. Accuracy versus the number of folds for random forest for a number of estimators equal 100.

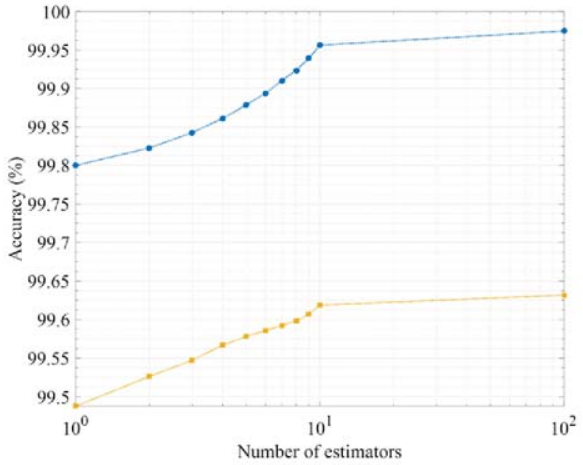


Fig. 3. Accuracy of the random forest for both training and testing with 100 estimators.

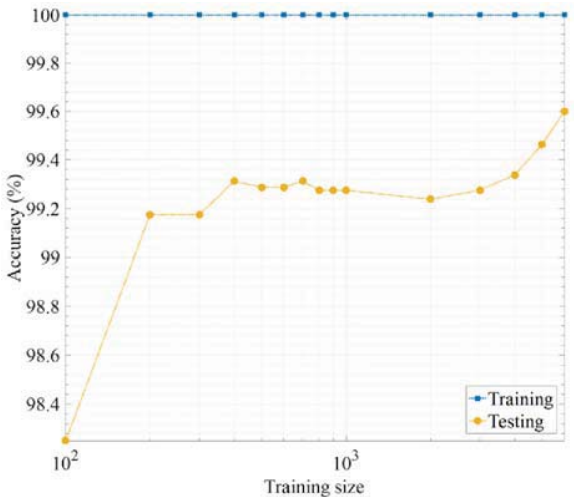


Fig.2. Accuracy versus number of estimators of random forest

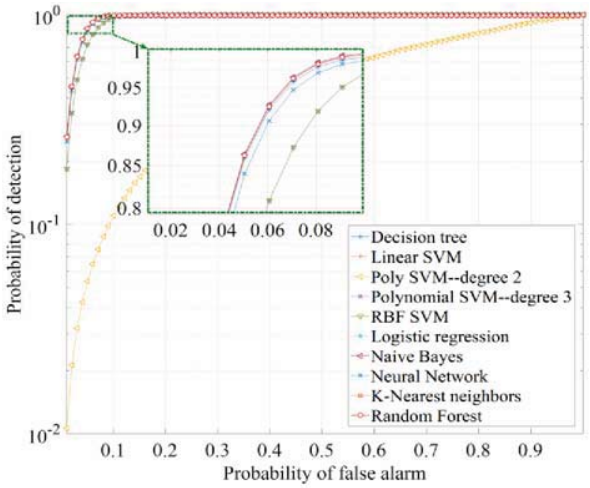


Fig. 4. Probability of detection versus the probability of false alarm

SVM are compared also based on the four metrics. Among all of these kernels, SVM linear and RF have very high probabilities of detection equal to 99.09% and 99.37%, respectively. However, for the probability of false alarm and probability of miss-detection, SVM linear has higher values than random forest. The table also shows that random forest provides the best results for every performance evaluation with an accuracy as high as 99.65%.

TABLE I. COMPARISON OF MACHINE LEARNING TECHNIQUES

| Models | Pd(%) | Pm (%) | Pfa(%) | A(%) |
|---|---|---|---|---|
| Linear SVM | 99.09 | 0.909 | 0.91 | 98.9 |
| Quadratic SVM | 4.05 | 95.9 | 95.95 | 51.8 |
| Cubic SVM | 82.5 | 17.53 | 17.5 | 91.5 |
| RBF SVM | 99.1 | 0.91 | 0.90 | 98.8 |
| Logistic Regression | 98.98 | 1.01 | 1.02 | 98.8 |
| Random Forest | 99.37 | 0.62 | 0.63 | 99.65 |
| KNN | 98.88 | 1.12 | 1.12 | 99.3 |
| Naïve Bayes | 98.6 | 1.43 | 1.4 | 99.27 |

## IV. CONCLUSION

Cognitive radio to a large extent will be adopted to manage the radio spectrum so that anyone could broadcast anything, on whatever frequency was most convenient, without interfering with anyone else's signal. Spectrum sensing is a core function of this technology. In this paper, we proposed a machine learning based scheme for spectrum sensing. This model performs spectrum sensing to determine the availability of frequency channels using machine learning classifiers. The spectrum sensing is modeled rigorously and a large-scale comprehensive dataset is built and used to train, validate, and test several machine learning techniques. These techniques are random forest, support vector machine with different kernels, decision tree, Naïve Bayes, K-nearest neighbors, and logistic regression. All models are extensively tested and evaluated using metrics such as the probabilities of detection, false alarm, and miss-detection as well as the accuracy of the classification. The simulation results show that the random forest outperforms all other machine learning methods.

## REFERENCES


[1] S. Haykin; D. J. Thomson and J. H. Reed, "Spectrum Sensing for Cognitive Radio", Proceedings of the IEEE. Vol. 97, No 5, pp. 849-877, 2009.

[2] Y. Zeng, and Y.-C. Liang, "Spectrum-Sensing Algorithms for Cognitive Radio Based on Statistical Covariances," IEEE Transactions On Vehicular Technology, Vol. 58, No. 4, May 2009

[3] Y. Arjoune and N. Kaabouch, "A Comprehensive Survey on Spectrum Sensing in Cognitive Radio Networks: Recent Advances, New Challenges, and Future Research Directions," Sensors 2019, 19, 126.

[4] S. Bhattarai, J. Park, B. Gao, K. Bian, and W. Lehr, "An Overview of Dynamic Spectrum Sharing: Ongoing Initiatives, Challenges, and a Roadmap for Future Research," IEEE Transactions On Cognitive Communications and Networking, Vol. 2, No. 2, June 2016

[5] X. Hong, J. Wang, C. Wang, J. Shi, Cognitive radio in 5G: a perspective on energy-spectral efficiency trade-off, IEEE Commun. Mag. 52 (7) (2014) 46–53.

[6] H. Reyes, S. Subramaniam, N. Kaabouch, and W. C. Hu. "A Bayesian inference method for estimating the channel occupancy." IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 1-6., 2016.

[7] I. F. Akyildiz, Lee, W.Y., Vuran, M.C. and Mohanty, S. "A Survey on Spectrum Management in Cognitive Radio Networks", IEEE Communications Magazine, 46 (4) (2008), pp. 40-48

[8] T. Yucek and H. Arslan "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications," IEEE Communications Surveys & Tutorials, Vol. 11, No. 1, pp.116-130, 2009

[9] A. Ali, and W. Hamouda, "Advances on Spectrum Sensing for Cognitive Radio Networks: Theory and Applications," IEEE Communications Surveys & Tutorials, Vol. 19, No. 2, pp. 1277-1303, 2017

[10] Y. Arjoune, Z. El Mrabet, H. El Ghazi, A. Tamtaoui, "Spectrum sensing: Enhanced energy detection technique based on noise measurement," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1-7, 2018.

[11] M. Naraghi-Pour, and T. Ikuma,"Autocorrelation-Based Spectrum Sensing for Cognitive Radios," IEEE Transactions On Vehicular Technology, Vol. 59, No. 2, pp. 718-733, February 2010

[12] A. Eftekhari, A. J. Romberg and M.B. Wakin, "Matched Filtering from Limited Frequency Samples", IEEE Transactions on Information Theory, vol. 59, 2013, pp. 3475-3496.

[13] H. Reyes, Hector, S. Subramaniam, N. Kaabouch, and W. C. Hu. "A Bayesian inference method for estimating the channel occupancy." Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 1-6. IEEE, 2016.

[14] R. Umar and A. U. H. Sheikh, "A Comparative Study of Spectrum Awareness Techniques for Cognitive Radio Oriented Wireless Networks", Physical Communication, 23 pages, 2012.

[15] K. Madushan, T. Kae, W. Choi, N. Saquib, and E. Hossain, "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks", IEEE Journal on selected areas in communications, vol. 31, no. 11, pp. 2209-2221, 2013.

[16] K. Zhang, J. Li, and F. Gao, "Machine Learning Techniques for Spectrum Sensing When Primary User Has Multiple Transmit Powers", IEEE Conference on Communication Systems, pp. 137-141, 2014.

[17] B. Khalfi, A. Zaid, B. Hamdaoui, "When Machine Learning Meets Compressive Sampling for Wideband Spectrum Sensing", Wireless Communications and Mobile Computing Conference, pp. 1120-1125, 2017.

[18] A. M. Mikaeil, B. Guo, Z. Wang, "Machine Learning to Data Fusion Approach for Cooperative Spectrum Sensing," Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 429-434, 2014.

[19] Y. Lu, P. Zhu, D. Wang and M. Fattouche, "Machine Learning Techniques with Probability Vector for Cooperative Spectrum Sensing in Cognitive Radio Networks," IEEE Wireless Communications and Networking Conference, pp. 1-6, 2016.

[20] Z. El Mrabet et al., "Primary User Emulation Attacks: A Detection Technique Based on Kalman Filter," J. Sens. Actuator Netw., vol. 7, no. 3, p. 26, Jul. 2018.

[21] S. El Barrak, A. Lyhyaoui, A.E. Gonnouni, A. Puliafito, S. Serrano, "SVM-MUSIC Algorithm for Spectrum Sensing in Cognitive Radio Ad-Hoc Networks." Ad-hoc, Mobile, and Wireless Networks. ADHOC-NOW 2017, vol 10517. Springer, Cham, 2017.