

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Data Mining Applied to Cognitive Radio Systems

Lilian Freitas, Yomara Pires, Jefferson Morais, João Costa and Aldebaro Klautau

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/51824>

1. Introduction

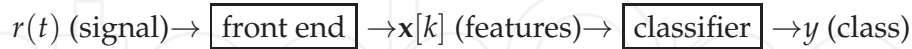
Cognitive radio (CR) is a novel technology that allows to improve spectrum utilization by enabling opportunistic access to the licensed spectrum band by unlicensed users [2]. This is accomplished through heterogeneous architectures and techniques of dynamic spectrum access. The CR is defined as an intelligent wireless communication system that is aware of its environment and is capable to learn from the environment and adapt its transmission parameters, such as frequency, modulation, transmission power and communication protocols [14].

An important aspect of a cognitive radio is spectrum sensing [10], which involves two main tasks: signal detection and modulation classification. Signal detection refers to detection of unused spectrum (spectrum holes). It is a simpler task and can be done, for example, by comparing the energy in the frequency band of interest with a predetermined threshold. This task is important so that the unlicensed users do not cause interference to licensed users. Modulation classification consists in automatically identifying the modulation scheme (PSK, FM, QAM, etc) of a given communication system with a high probability of success and in a short period of time. The identification of the modulation scheme allows the cognitive radio to demodulate the received signal. In order to accomplish the task of modulation classification, several data mining techniques can be applied, such as artificial neural networks, support vector machine, Bayesian classifiers, etc.

This chapter aims to evaluate different algorithms for classification of modulation signals on spectrum sensing. The features used for classification are based on a well-established technique called cyclostationarity [7, 10]. Based on these features are evaluated the performances of five data mining techniques: naïve bayes, decision tree, k-nearest neighbor (KNN), support vector machine (SVM) and artificial neural networks (ANN). The choice of such techniques was based on the fact that they are the most popular representatives of different learning paradigms.

2. The problem of modulation classification

A modulation classification system consists of a *front end* and a *back end* or classifier. The *front end* converts the received signal $r(t)$ to a vector $\mathbf{x}[k], k = 1, \dots, N$ composed of N elements. Having $\mathbf{x}[k]$ as input, the classifier decides the class $y \in \{1, \dots, C\}$ among C pre-determined modulation schemes. The process is depicted in the diagram below:



The feature selection is a key step in the performance of the classifier. This selection depends on factors such as the modulation type to be classified, the signal to noise ratio, the presence of fading, the frequency offset, etc. This chapter uses the cyclostationarity to extract features of modulation due to its reduced sensibility to noise and interfering signals, and also its ability to extract signal parameters such as the carrier frequency and the symbol rate [7].

In the literature there are numerous works that combine different techniques of extraction features and classifiers to perform the modulation classification, as shown in Table 1.

Ref.	front end	Classifier	Noise/ Interference	SNR (dB) (min:Δ:max)	Modulations	Training/ Testing
[15]	CSS	SVM	AWGN/ freq. offset	-5:2:10	BPSK, 4PAM, 16QAM, 8PSK	500/500
[16]	features derived from amplitude, frequency and phase	neural networks	AWGN	-5:5:20	2ASK, 4ASK, 2FSK, BPSK, QPSK, AM, DSB, SSB, FM OFDM, 16QAM, 64QAM	1200/1200
[18]	cyclic spectrum	neural network and HMM	AWGN	-10:5:10	BPSK, QPSK/QAM, FSK, MSK	several
[13]	spectrogram time-frequency	rule based	AWGN	0:2:12	ASK, FSK, PSK	NE/400
[4]	linear transform of amplitude and phase	joint moments	AWGN/ freq. offset	0:1:5	BPSK, QPSK	NE/100
[20]	wavelet transform	decision threshold	AWGN	0:5:15	FSK, PSK, QAM	NE/1000
[19]	fourth-order cumulant	decision threshold	AWGN/ freq. offset	0:5:20	BPSK, QPSK, 8PSK pi/4 DQPSK	NE/100
[24]	forth-order and sixth-order cumulants	ARBF	not informed	0:5:20	4ASK, 2ASK, 2PSK 4PSK, 16QAM	50/50
[1]	cyclostationarity	neural networks	AWGN/ freq. offset	-10:5:15	BPSK, QPSK FSK, MSK, AM	NE/1000
[11]	Renyi entropy, high order statistics	SVM	AWGN	10 dB	AM, FM, AM-FM QPSK	200/200

Table 1. Examples of different *front end* and *classifier* used in the literature. NA: not available.

These works show good results in the experimental setup in which they were assessed. However, these works are evaluated on different operating conditions (signal-to-noise ratio, type of noise and distortion) and use different modulations. Thus, it is difficult to directly compare the results, or even to reproduce the results presented.

In this chapter, the comparison of main classifiers available in the literature is performed, considering the same operating conditions: five modulation schemes (AM, BPSK, QPSK, BFSK and 16QAM); channel with additive white Gaussian noise (AWGN) and Rayleigh multipath fading; all the modulated signals adopt the same symbol rate, sampling frequency and carrier

frequency. For every modulation scheme 1500 samples are generated under each SNR (from -10 dB to 10 dB at intervals of 5 dB), in which 750 samples are used for training, and the other 750 samples for testing.

3. Front end: Cyclostationarity

Cyclostationarity is a technique that extracts features of the signals. Signals are characterized as cyclostationary since their mean and autocorrelation are periodic with some period T . This is

$$\begin{aligned} M_x(t+T) &= M_x(t) \\ R_x(t+T, u+T) &= R_x(t, u) \end{aligned}$$

for all t and u , where $x(t)$ is a signal said to be cyclostationary.

Modulated signals are cyclostationary since they are coupled with several sources of periodicities such as sine wave carriers, pulse trains, repeating spreading, hopping sequences, or cyclic prefixes. These introduced periodicities cause spectral redundancy, which can be measured by the correlation between spectral components of cyclostationary signals. This periodicity appearing in transmitted signal of the users can be used by cognitive radio to detect and identify user signals.

Since the autocorrelation function $R_x(\tau)$ of received signal $x(t)$ is periodic, it can be represented by a Fourier Series, shown in Equation 1,

$$R_x(t + \tau/2, t - \tau/2) = \sum_{\alpha} R_x^{\alpha}(\tau) e^{i2\pi\alpha t} \quad (1)$$

where $R_x^{\alpha}(\tau)$ is the Fourier Series coefficients called *cyclic autocorrelation function* with spectral components at cyclic frequencies α . The Fourier coefficients may be obtained by Equation 2.

$$R_x^{\alpha}(\tau) \triangleq \frac{1}{T} \int_{-T/2}^{T/2} R_x(t + \frac{\tau}{2}, t - \frac{\tau}{2}) e^{-i2\pi\alpha t} dt, \quad (2)$$

The density of spectral correlation is the Fourier Transform of cyclic autocorrelation function $R_x^{\alpha}(\tau)$ and *spectral correlation function* (SCF), $S_x^{\alpha}(f)$, is the density of correlation between spectral components at $(f + \alpha/2)$ and $(f - \alpha/2)$, and is given by Equation 3,

$$S_x^{\alpha}(f) = \lim_{T \rightarrow \infty} \lim_{\Delta t \rightarrow \infty} \frac{1}{T\Delta t} \int_{-\Delta t/2}^{\Delta t/2} X_T(t, f + \frac{\alpha}{2}) X_T^*(t, f - \frac{\alpha}{2}) dt \quad (3)$$

where $X_T(t, f)$ is the spectral components of received signal $x(t)$ at frequency f with bandwidth $1/T$ as defined in Equation 4.

$$X_T(t, f) = \int_{t-T/2}^{t+T/2} x(u) e^{-i2\pi fu} du \quad (4)$$

The SCF is a three-dimensional function; therefore, to reduce the calculations for the classifier, it is possible to use the peak values of normalized SCF as features to distinguish each

modulation, that is, the *cyclic domain profile* (CDP), obtained by Equation 5.

$$I(\alpha) = \max_k |S_x^\alpha(f)|. \tag{5}$$

In order to illustrate the use of the cyclostationarity technique, Figure 1 shows the estimation of the normalized SCF for BPSK and QPSK modulations respectively. The Figure 2 shows the cyclic domain profile for the BPSK and QPSK modulations. These examples adopted a sampling frequency $f_s = 8192$ Hz, carrier frequency $K = 2048$ Hz, cyclic frequency resolution $\Delta\alpha = 20$ Hz and frequency resolution $\Delta f = 80$ Hz.

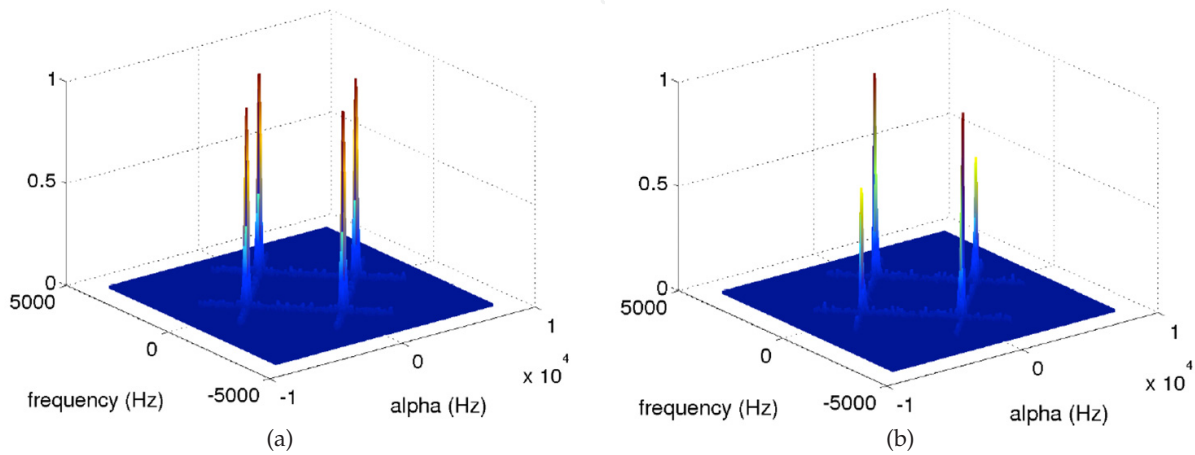


Figure 1. Spectral correlation function. (a) BPSK. (b) QPSK

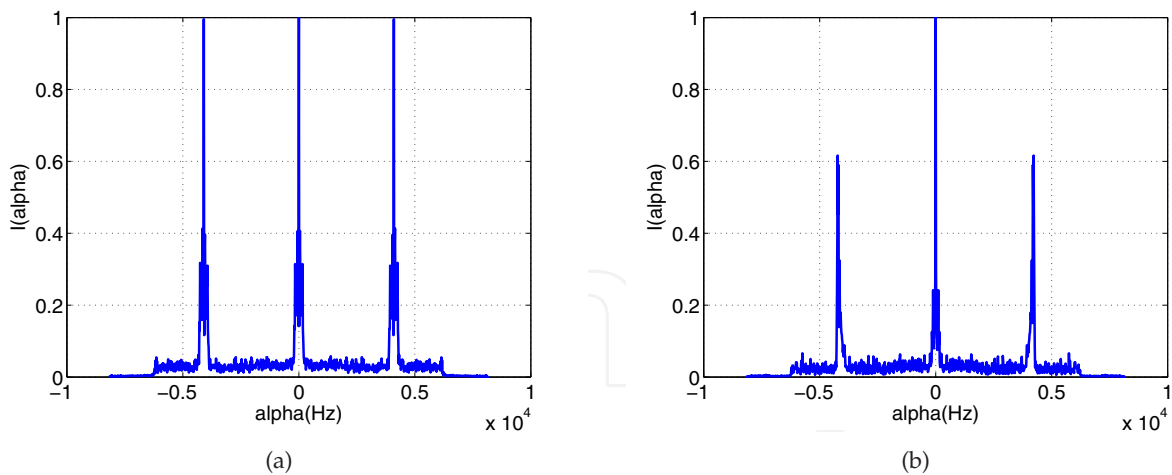


Figure 2. Cyclic domain profile. (a) BPSK. (b) QPSK

Note that different modulations have different CDP, thus these features were used as input to the block classifier. The cyclostationary features of modulated signals have been increasingly considered for use in a large range of applications, including signal detection, classification, synchronization and equalization. Its main advantages are the reduced sensibility to noise

and interfering signals, and also its ability to extract signal parameters such as the carrier frequency and the symbol rate [7].

4. Classifiers

4.1. Naïve Bayes

The naïve Bayes classifier is based on Bayes' theorem. This classifier is particularly useful when the input data dimensionality is high. Thus, to represent the classifier in the cognitive radio system, we adopt the nomenclature used in [6], where $P(y|\mathbf{x})$, $P(\mathbf{x}|y)$, $P(y)$ and $P(\mathbf{x})$ are called posterior, likelihood, prior and evidence, respectively, and are related through Bayes' rule,

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}. \quad (6)$$

This classifier attempts to select the label

$$\mathcal{F}(\mathbf{x}) = \arg \max_{y=1,\dots,Y} P(\mathbf{x}|y)P(y), \quad (7)$$

which maximizes the posterior probability. However, neither $P(y)$ nor $P(\mathbf{x}|y)$ is known. Hence, the classifiers use estimates $\hat{P}(y)$ and $\hat{P}(\mathbf{x}|y)$ and maximize

$$\mathcal{F}(\mathbf{x}) = \arg \max_{y=1,\dots,Y} \hat{P}(\mathbf{x}|y)\hat{P}(y). \quad (8)$$

In most cases, the prior $P(y)$ can be reliably estimated by counting the labels in the training set, i.e., we assume that $\hat{P}(y) = P(y)$. In order to estimate $\hat{P}(\mathbf{x}|y)$ is often the most difficult task. Hence, Bayes classifiers typically assume a parametric distribution $\hat{P}(\mathbf{x}|y) = \hat{P}_{\theta_y}(\mathbf{x}|y)$ where θ_y describes the distribution's parameters to be determined (e.g., the mean and covariance matrix if the likelihood model is a Gaussian distribution).

The naïve Bayes algorithm assumes that the attributes (x_1, \dots, x_K) of \mathbf{x} are conditionally independent of each other, given y . It means that the algorithm simplifies the representation of $P(\mathbf{x}|y)$, and the estimation problem from the training set. Whereas. In the case where $\mathbf{x} = (x_1, x_2)$, we have:

$$P(\mathbf{x}|y) = P(x_1, x_2|y) = P(x_1|x_2, y)P(x_2|y) = P(x_1|y)P(x_2|y) \quad (9)$$

where $P(x_1, x_2|y) = P(x_1|x_2, y)P(x_2|y)$ is a general property from conditional probability definition, while $P(x_1, x_2|y) = P(x_1|y)P(x_2|y)$ is only valid for conditional independence. Generalizing Equation 9, we have:

$$P(\mathbf{x}|y) = P(x_1, \dots, x_K|y) = \prod_{i=1}^K P(x_i|y). \quad (10)$$

When training a naïve Bayes classifier, this will produce a probability distribution $P(\mathbf{x}_i|y)$ and $P(y)$ for all values of y , i.e, $y_k, k = 1, \dots, Y$. To calculate the posterior probability of each class

y , we use Bayes' theorem:

$$p(y_k|\mathbf{x}) = \frac{P(y_k)P(x_1, \dots, x_K|y_k)}{\sum_j P(y_j)P(x_1, \dots, x_K|y_j)} \quad (11)$$

Assuming x_i is conditionally independent given y , we can rewrite Equation 10 as:

$$p(y_k|\mathbf{x}) = \frac{P(y_k) \prod_i P(x_i|y_k)}{\sum_j (P(y_j) \prod_i P(x_i|y_j))}. \quad (12)$$

Equation 12 is the fundamental equation of a naïve Bayes classifier. Given a new sample \mathbf{x} , this equation shows how to calculate the probability for each y . Such calculation depends only on observed attribute values and distributions $P(y)$ and $P(x_i|y)$ estimated from the data training. If it is desired only to the most likely value of y , then we can simplify to

$$\mathcal{F}(\mathbf{x}) = \arg \max_{y_k} P(y_k) \prod_i P(x_i|y_k) \quad (13)$$

or, using the fact that the logarithm is a monotonic function:

$$\mathcal{F}(\mathbf{x}) = \arg \max_{y_k} \left[\log P(y_k) + \sum_i \log P(x_i|y_k) \right]. \quad (14)$$

4.2. Decision tree

A decision tree is a model of predictive machine learning which performs the decision of a new instance based on the value of its various attributes [23]. It consists of a structure where leaf nodes represent tests of one or more attributes. The branches of these nodes are the possible values of these attributes. The terminal nodes are the result of classification. In order to perform the classification of a new instance, a decision tree is created based on the values of the attributes of the training set. This chapter uses the decision tree implemented in the Weka software, called J4.8, which is an implementation of the C4.5 algorithm, which was developed by J. Quinlan [17] and probably the most famous algorithm for the design of decision trees.

A decision tree is formed by a set of classification rules. Each path from the root to a leaf represents one of these rules. The decision tree should be set so that for each observation in the database, there is only one path from root to leaf. Classification rules are composed of an antecedent (precondition) and a consequent (conclusion). An antecedent should be formed by one or more predictive attributes, while the consequent defines the class or classes.

A key issue for building a decision tree is the strategy for the choice of features that can determine the class to which a *sample* belongs. Measures based on entropy are commonly used to address this problem, which measures the randomness of the value of a feature before deciding which feature to use to predict the class.

Decision trees are methods that use a recursive algorithm for successive divisions in a training set. The main problem is then the reliability of estimates of the error used to select the divisions. Despite the fact that estimate obtained with the training data used during the

growth of the tree known as "resubstitution error" continues to decrease, generally, the choices of the division in higher levels of the tree does not produce very reliable statistics. Therefore, the quality of the sample directly influences the accuracy of the estimates of the error. Since each iteration of the algorithm divides the set of training data, the internal nodes make decisions from ever smaller samples. This means that the error estimates are less reliable as the tree grows. Thus, *pruning methods* have been used to minimize this problem and avoid *overfitting* [6, 23].

Basically, there are two classes of methods in pruning a decision tree: a post-pruning and pre-pruning. In this chapter the *post-prune* method is used, which consists in allowing the tree to grow to a maximum size, i.e., until the leaf nodes that have minimal impurity, for subsequent application of the pruning.

4.3. SVM classifier

A support vector machine (SVM) is a class of learning algorithms based on statistical learning theory, which implements the principle of structural risk minimization [21]. The goal of an SVM classifier is to find a maximum margin hyperplane in a feature space. A hyperplane function is to be a decision surface such that the margin of separation between examples of one class and another is at a maximum [5].

More specifically, a SVM is a binary classifier given by

$$f(\mathbf{x}) = \sum_{m=1}^M \alpha_m \mathcal{K}(\mathbf{x}, \mathbf{x}_m) + c,$$

where $\mathcal{K}(\mathbf{x}, \mathbf{x}_m)$ is the kernel function between the test vector \mathbf{x} and the m -th training example \mathbf{x}_m , with $c, \alpha_m \in \mathbb{R}$. The effectively used examples have $\alpha_m \neq 0$ and are called *support vectors*. In the literature, several possibilities of kernels are presented in applications involving pattern recognition, such as linear, Gaussian, polynomial, sigmoid and radial basis functions.

A SVM with a linear kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}_m) = \langle \mathbf{x}, \mathbf{x}_m \rangle$ given by the inner product between \mathbf{x} and \mathbf{x}_m can be converted to a perceptron $f(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x} \rangle + c$, where $\mathbf{a} = \sum_{m=1}^M \alpha_m \mathbf{x}_m$ is pre-computed.

Therefore, linear SVMs were adopted in this chapter due to their lower computational cost when compared to non-linear SVMs with kernels such as the Gaussian [5]. To combine the binary SVMs $f_b(\mathbf{x}), b = 1, \dots, B$, to obtain $F(\mathbf{x})$ this work adopted the *all-pairs* error-correcting output code (ECOC) matrix with Hamming decoding [3], where the winner class is the one with the majority of "votes". Note that an alternative to all-pairs, which uses $B = 0.5C(C - 1)$ SVMs, is the one-vs-all ECOC that uses $B = C$ SVMs [12].

4.4. Artificial neural networks

Artificial neural networks (ANN) are parallel distributed systems composed of simple processing units called *neurons* that compute some mathematical function, usually nonlinear. Such units are arranged in one or more layers and interconnected by so-called synaptic

weights. The intelligent behavior of ANN comes from the interactions between the processing units of the network.

A neuron consists of a sum of weights and inputs, and an activation function. The weight of the connections are set by a rule of training, according to the patterns presented. In this chapter, a neural network was used called multilayer perceptrons with the backpropagation algorithm for training, which has shown good results in classification problems.

The algorithm multilayer perceptron backpropagation (also called the generalized Delta Rule) consists in a process of supervised learning using a predetermined set of pairs of input and output to adjust the weights in the network using an error correction scheme held in propagation cycles [6, 9]. The backpropagation is divided into two phases: the *first step* is forward the input vector from the first through the last layer and to compare the output value to the desired value. The *second phase* consists of backwarding the error based on the last layer through the input layer by adjusting the weights of the neurons of the hidden layers. After adjusting all the weights of network, is given one more set of examples is given, ending a epoch. This process is repeated until the error is acceptable for the training set, referred to as the convergence time of the network.

The performance of a multilayer perceptron neural network during training depends on the following parameters [9]:

- *Initialization of weights.* The weights of the connections between neurons can be initialized randomly or uniformly.
- *Learning rate.* The learning rate controls the speed of learning, increasing or decreasing the set of weights performed at each iteration during training. Intuitively, its value must be greater than 0 and less than 1. If the learning rate is too small, learning will take place very slowly. Where the rate is very large (greater than 1), the correction would be greater than the observed error, causing the neural network learning point exceeding its greatest value, making the training process unstable.
- *Transfer function parametrization.* Also known as threshold logic, this function is the one which defines and sends out the value passed by the neuron activation function. The activation function can take many forms and methods. The best known are the following: linear function, sigmoid function and exponential function.

4.5. KNN

The classifiers that simply store the training data are called “lazy” classifiers, or known as IBL (instance based learning) [23]. The k-nearest neighbor (KNN) is a method of this family and stores examples in memory as points in n -dimensional space defined by n attributes that describe the examples [6]. Thus, for each new example to classify, KNN uses the training data to determine the examples in the database that are “nearest” to the example in the analysis. With each new example to be classified, a sweep in the training data, is made, which causes a large computational effort.

Suppose a training set with N examples. Let $\mathbf{x} = (x_1, \dots, x_k)$ be a new example, not yet classified. In order to classify it, calculate distances by a measure of similarity between \mathbf{x} and all examples in the training set and consider the K closest examples (with the lowest distance) for \mathbf{x} . The example \mathbf{x} is classified according to the most frequent class y among the K examples found.

The distance between two examples is calculated by a measure of similarity. A popular measure of similarity is the Euclidian distance [6]. This measure calculates the square root of the sum of the squares of the differences between the vectors \mathbf{x} and $\hat{\mathbf{x}}$:

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\sum_{i=1}^K (x_i - \hat{x}_i)^2} \quad (15)$$

This chapter uses the Euclidean distance. Based on this metric, the KNN searches the “nearest neighbors” to classify new examples.

5. Results

The simulations aim to evaluate the reliability of cyclostationarity technique for feature extraction and to compare the performance of data mining techniques like naïve bayes, decision tree, KNN, SVM and ANN, under various conditions. The signals were modulated using AM, BPSK, QPSK, BFSK and 16-QAM modulations. The signals were propagated through two types of channel models: AWGN channel and an AWGN and multipath fading channel. The signal-to-noise ratio (SNR) was varied randomly from -10 to 10 dB as part of the simulation. In both channel models, carrier frequency $f_c = 2.4$ GHz, sampling period $T_s = 0.167$ ns, square-root raised cosine with roll-off factor $r = 0.1$, number of symbols $N_{symbol} = 100$, FFT points $N_{fft} = 512$ was used.

In order to implementation of the classifiers we used the WEKA software [22], which is a collection of machine learning algorithms for data mining tasks. Weka is open source software issued under the General Public License. We adopted the following settings for the classifiers:

- **Naïve Bayes:** The naïve Bayes used a normal distribution for numeric attributes, and the parameter K (*UseKernelEstimator*) set to False, which corresponds to the standard naive Bayes.
- **KNN:** In the configuration we adopted the KNN search algorithm of neighbors based on Euclidean distance.
- **J4.8:** The J4.8 has been configured with automatic selection for the confidence factor parameter C . Thus, the results presented represent the best result achieved for confidence factor values which varied between [0.1, 0.25 and 0.5];
- **SVM:** The SVM has been configured with linear kernel ($K = 0$), with the cost parameter C ranging from [2, 1, 0.5 and 0.25] and degree of the kernel $D = 3$.

- **ANN:** We adopted a neural network multilayer perceptron with learning algorithm backpropagation, the number of neurons in the hidden layer ranging from [60, 110, 130 and 160] neurons, learning rate ranging between a rate of [0.1, 0.5 and 0.9], and time varying between [0.1, 0.2 and 0.4].

5.1. Sample complexity

The first experiment aimed to analyze the accuracy of classifiers with the variation in the number of samples used in the training phase. This analysis was performed using the sample complexity curves [8].

The sample complexity curve aims to determine how many samples are required for the classifier to achieve a certain level of performance. The abscissa represents the number of samples in the training set and the ordinate represents the percentage of correct classifications obtained in the test phase. It should be noted that only the number of samples of the training set varies, while the number of samples in the test phase remains fixed.

The classifiers were trained with different numbers of samples, varying between [50, 150, 300, 450, 750 and 1000] samples of each modulation. In the test phase, 1000 samples were used for each modulation. Figure 3 through Figure 6 show the results obtained for a multipath fading channel, configured with Doppler frequency $FD = 50$ Hz and AWGN.

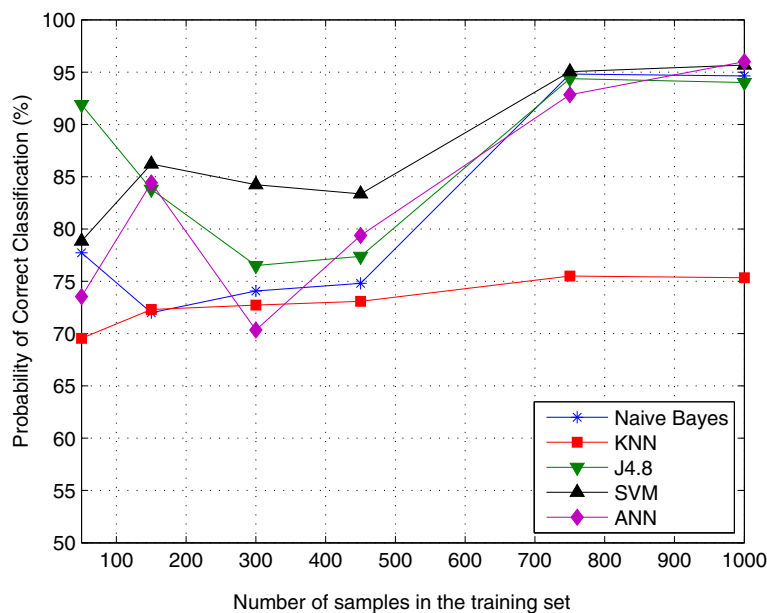


Figure 3. Sample complexity for SNR = -10 dB.

Through an analysis of sample complexity curves, it was decided to work with 750 training samples of each modulation, since this number had a good performance for the different classifiers which were evaluated.

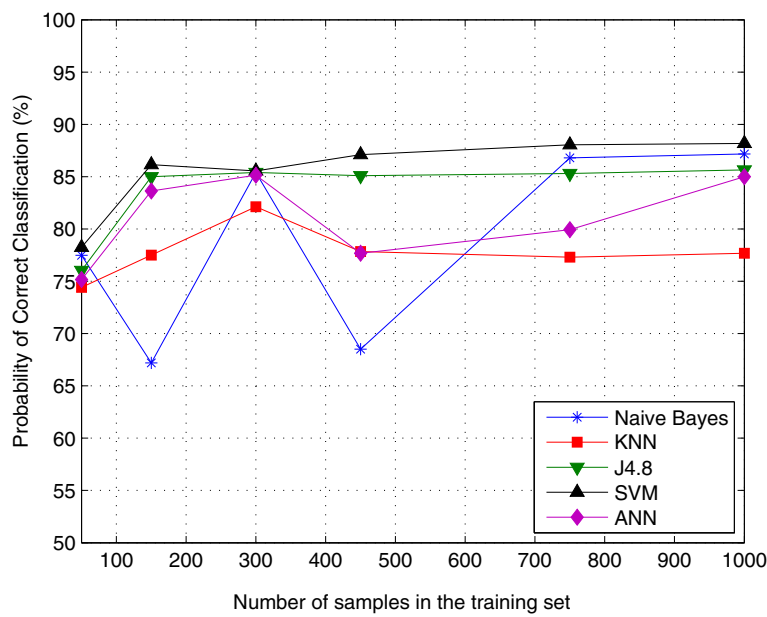


Figure 4. Sample complexity for SNR = -5 dB.

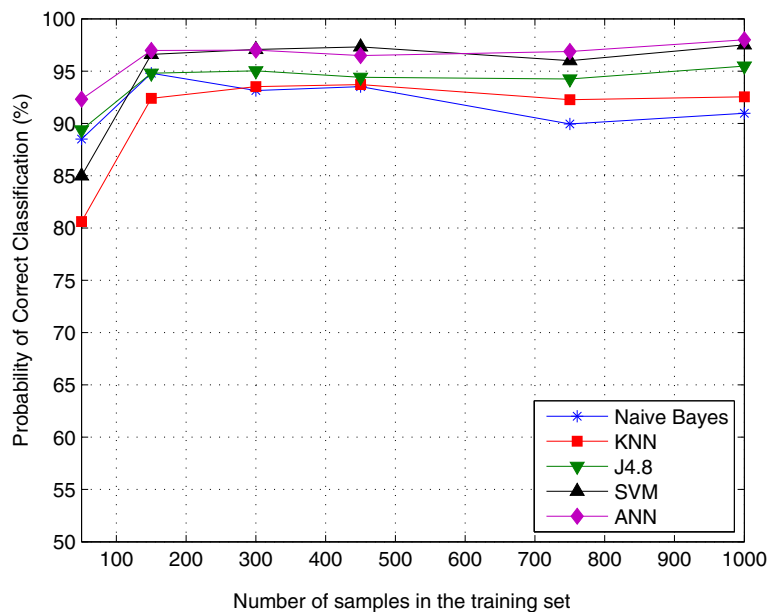


Figure 5. Sample complexity for SNR = 5 dB.

5.2. Simulation result of AWGN channel

In this scenario, the SNR is varied from a range of -15 to 15 dB range. The training, testing and validation sets were composed of 750 different samples of each modulation. The classifiers were trained and tested with the same values of SNR. Figure 7 shows the results for the classification of AM, BPSK, QPSK and BFSK modulations.

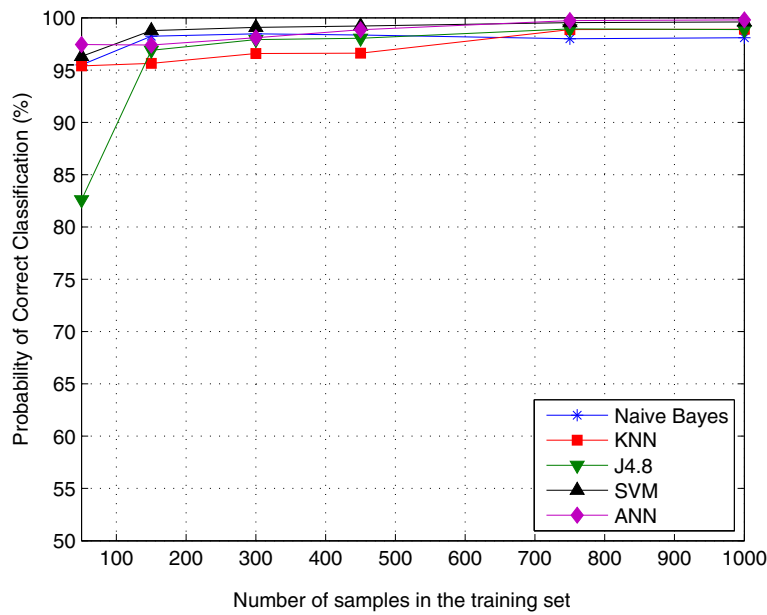


Figure 6. Sample complexity for SNR = 10 dB.

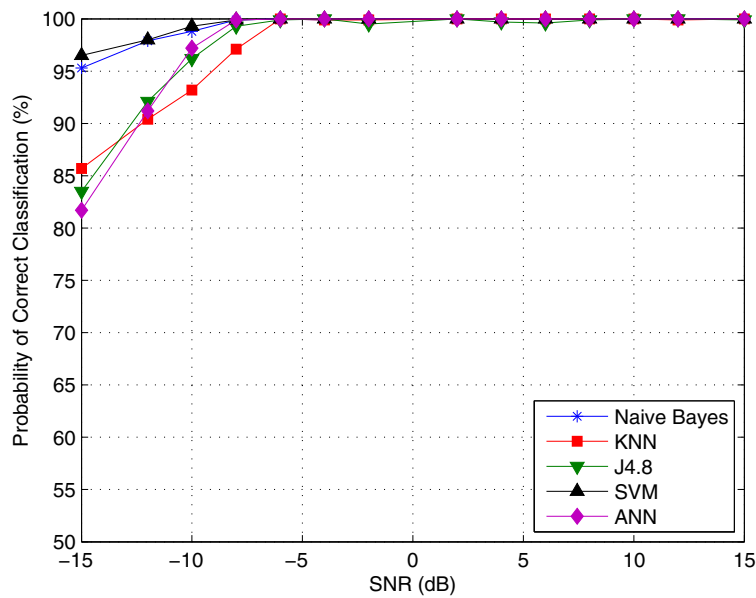


Figure 7. Performance of classifiers in a AWGN channel.

The results show that for SNR values greater than -5 dB, all classifiers presented excellent performance with nearly 100% correct classification. Among the evaluated classifiers, the SVM had a higher percentage of correct classification, even for SNR values lower than -5 dB.

Table 2 allows the analysis of the performance of classifiers in the worst case, i.e., SNR = -15 dB. The results show that the greater number of errors occurs in the classification of QPSK and BPSK modulations, mainly by KNN (with 14.3 % errors) and decision tree J4.8 (16.5 % errors) classifiers.

Modulations	N.	Bayes	J4.8	KNN	SVM	ANN
AM	100	100	100	100	100	100
BPSK	90.0	67.6	85.2	92.8	79.2	
QPSK	91.2	67.2	58.0	93.2	47.6	
BFSK	100	99.2	99.6	100	100	

Table 2. Performance of classifiers in SNR = -15 dB.

Table 3 shows the confusion matrix of the J4.8 classifier considering a SNR = -15 dB. It is observed that at low SNR, the classification error occurs to distinguish the QPSK and BPSK modulation, due to distortions in their features. Figure 8 shows the profiles of the BPSK and QPSK modulations in SNR = -15 dB and 15 dB.

Classifier as ->	AM	BPSK	QPSK	BFSK
AM	750	0	0	0
BPSK	3	507	237	3
QPSK	3	243	504	0
BFSK	6	0	0	744

Table 3. Confusion matrix of the J4.8 classifier, SNR = -15 dB.

In order to analyze the degree of generalization of the classifiers, two experiments were realized. In the first, the number of samples in training set was fixed at 750 samples for each modulation with SNR = 5 dB. The number of samples in the test set was varied with SNR values from -15 to 15 dB. The goal was to evaluate the performance of classifiers when tested with SNR values for which they were not trained. The results are shown in Figure 9. It is observed that SVM and ANN presented the best performance, which can be seen, for example, comparing the performance of classifiers in SNR = -5 dB.

The second experiment was to train classifiers with SNR values from -15 to 15 dB. Then the classifiers were tested with specific values of SNR, which are indicated on the axis of abscissa in Figure 9.

In this experiment, the performance of the classifiers obtained a considerable increase. The ANN and SVM classifiers presented the best performance; on the other hand, the naïve Bayes classifier had the worst performance.

Classifiers	Correct class. (%)
Naïve Bayes	89.1
KNN	91.9
J4.8	92.7
SVM	97.5
ANN	100

Table 4. Performance of classifiers when trained and tested with different SNR values.

In the literature there are studies indicating that the classification of QAM using cyclostationarity is difficult due to the fact that high-order QAM modulations do not exhibit periodicity of 2nd order, or in some cases, exhibit similar characteristics of QPSK

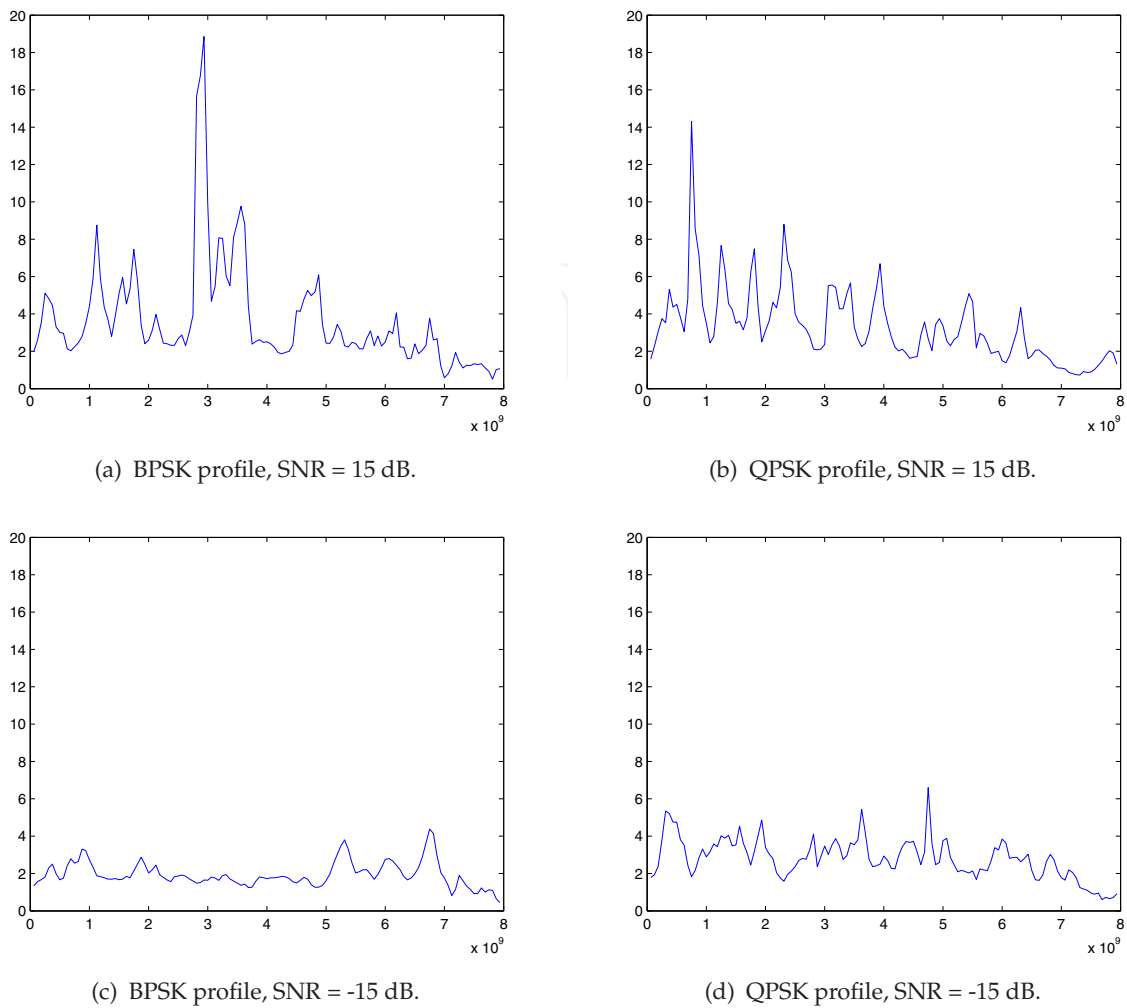


Figure 8. Profiles of the BPSK and QPSK modulations.

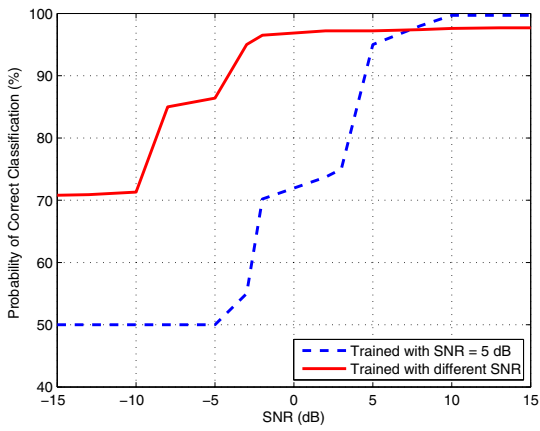
modulation [1]. The results that follow show the performance of classifiers to classify the 16-QAM modulation. Figure 10 allows comparison of the performance of classifiers when the 16-QAM modulation is included.

The results show that at low SNR, the performance of the classifiers decreases, when included the 16-QAM modulation. However, with increasing SNR, the performance of the classifiers is close to 100% of a correct classification.

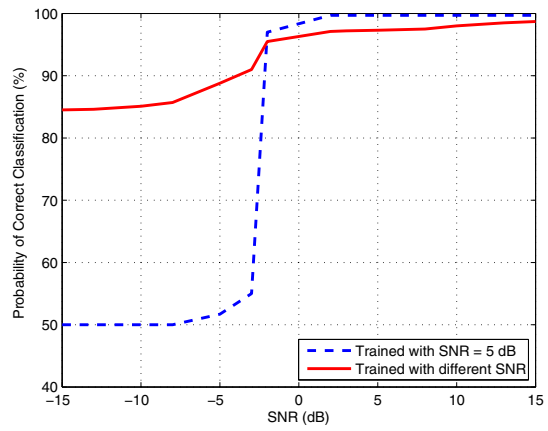
In general, the SVM classifier obtained the best results, which can be justified by their robustness, due to its mathematical formulation based on the search of the optimal solution. The naïve Bayes, despite being a simplistic method, also performed well, better than some classifiers already recognized as an ANN and KNN.

6. Simulation result of a multipath rayleigh fading and AWGN channel

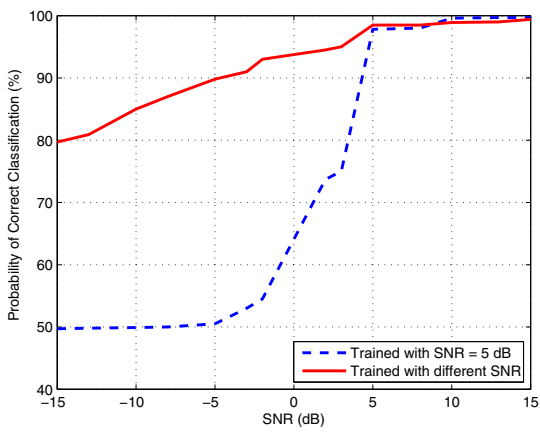
Figure 11 through Figure 13 show the results for SNR values from -15 to 15 dB and Doppler frequency $FD = [50, 150 \text{ and } 300]$ Hz. Modulations used were AM, BPSK, QPSK, BFSK and 16-QAM.



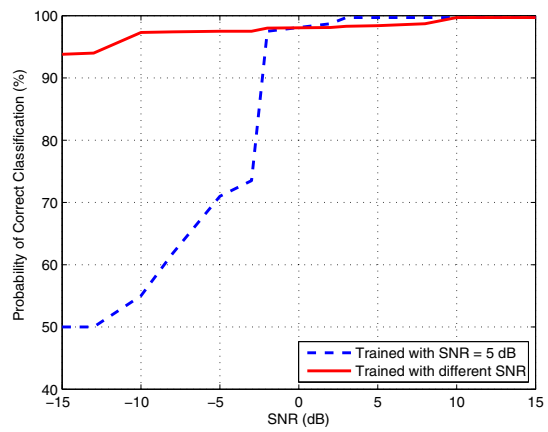
(a) Naïve Bayes



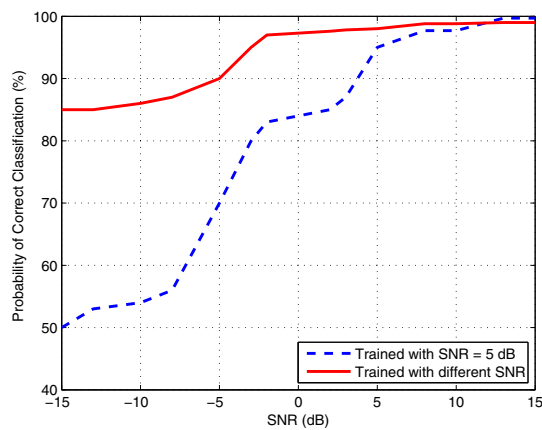
(b) KNN



(c) Decision tree J4.8

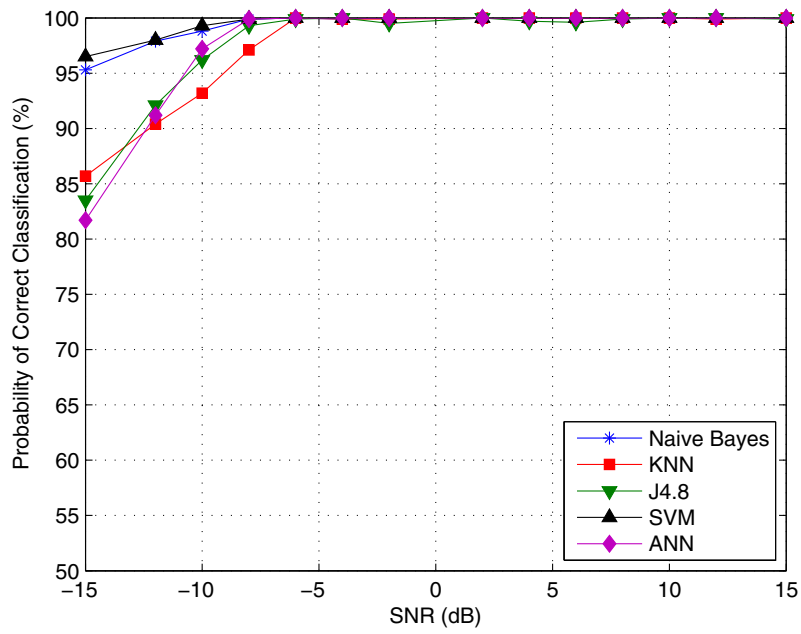


(d) SVM

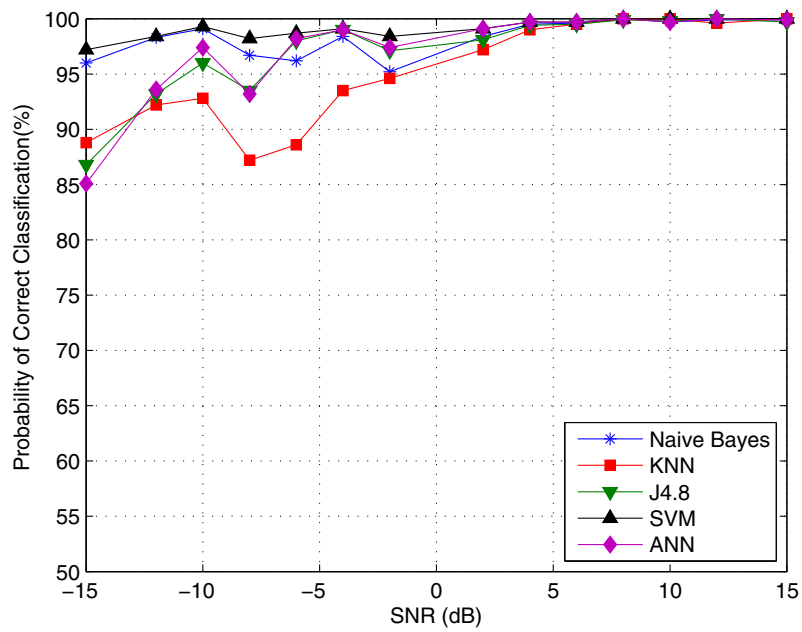


(e) ANN

Figure 9. Performance of classifiers when trained and tested with different SNR values. Abscissa indicates the SNR adopted for the test set.



(a) Without the 16-QAM modulation.



(b) With the 16-QAM modulation.

Figure 10. Performance of the classifiers for the classification of 16-QAM modulation.

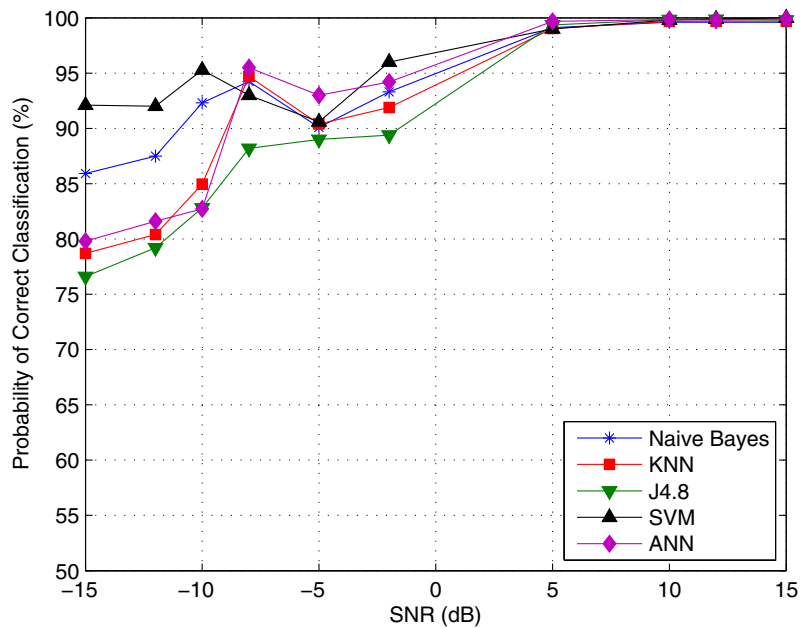


Figure 11. Performance of classifiers. Rayleigh fading channel, FD=50 Hz.

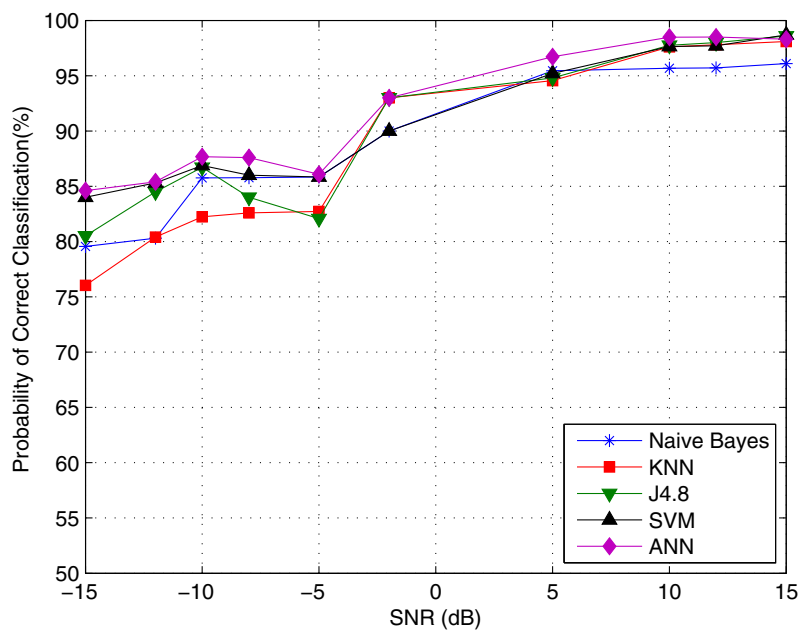


Figure 12. Performance of classifiers. Rayleigh fading channel, FD=150 Hz.

The results show that in general, all the classifiers had good performance. However, the decision tree KNN and J4.8 proved very susceptible to noise, especially for SNR values between -15 dB and 5 dB.

Furthermore, comparison between the Rayleigh and AWGN channels shows that there was a decrease in the performance of the classifiers. In the experiments, it was used a uniform procedure was used for selection of models of classifiers (i.e., not invested much in the tune

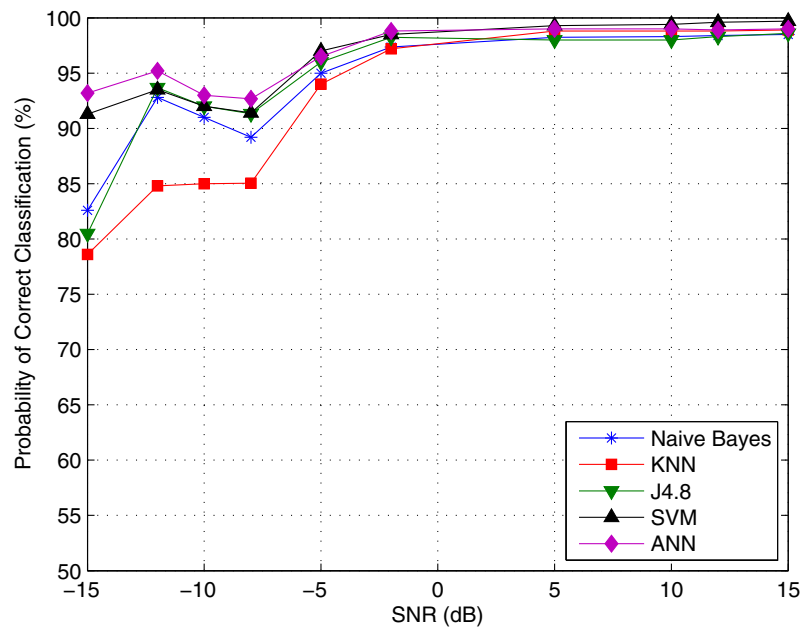


Figure 13. Performance of classifiers. Rayleigh fading channel, $FD=300$ Hz.

of a specific classifier). This may explain the variation in results for AWGN and multipath fading. A more detailed investigation about the parameters of classifiers such as SVM and ANN would probably improve the results.

7. Conclusions

This chapter discussed the task of modulation classification in cognitive radio. The modulation classification becomes fundamental, since this information allows the RC to adapt its transmission parameters for the spectrum to be shared efficiently, without causing interference to other users. A modulation classifier was implemented based on the characteristics of cyclostationarity of modulated signals. The performance of five data mining techniques were evaluated: naïve Bayes, decision tree J4.8, KNN, SVM, and ANN. In this evaluation, the signal classifications were performed to classifier AM, BPSK, BFSK, QPSK and 16-QAM modulations. An environment with multipath Rayleigh fading and AWGN was adopted.

Simulation results show that it is possible to classify the incoming signals, even at very low SNR, if the The cyclostationarity technique proved an effective technique for feature extraction, even in environments with low SNR. The SVM classifier with a linear kernel presented the best results, even in a fading multipath configuration.

The evaluation of algorithms for modulation classification proposed may serve as a starting point for researchers who want to compare results systematically.

Author details

Lilian C. Freitas and João Costa

Applied Electromagnetism Laboratory (LEA), Federal University of Pará (UFPA), Belém – PA – Brazil

Yomara Pires, Jefferson Morais and Aldebaro Klautau
Signal Processing Laboratory (LaPS), Federal University of Pará (UFPA), Belém – PA – Brazil

8. References

- [1] A. Fehske, J. Gaeddert, J. R. [2005]. A new approach to signal classification using spectral correlation and neural networks, *DySPAN* pp. 144–150.
- [2] Akyildiz, I., Lee, W., Vuran, M. C. & Mohanty, S. [2006]. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey, *Computer Networks: The International Journal of Computer and Telecommunications Networking* 50: 2127–2159.
- [3] Allwein, E., Schapire, R. & Singer, Y. [2000]. Reducing multiclass to binary: A unifying approach for margin classifiers, *Journal of Machine Learning Research* pp. 113–141.
- [4] D. Shimbo; I. Oka; S. Ata; [2007]. An improved algorithm of modulation classification for digital communication signals based on wavelet transform, *Radio and Wireless Symposium, 2007 IEEE* 03: 567–570.
- [5] Cristianini & Shawe-Taylor, J. [2000]. *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press.
- [6] Duda, R., Hart, P. & Stork, D. [2001]. *Pattern classification*, Wiley.
- [7] Gardner, W. A. & Spooner, C. M. [1992]. Signal interception: Performance advantages of cyclic-feature detectors, *IEEE Transactions on Communications* 40: 149–159.
- [8] Hastie, T., Tibshirani, R. & Friedman, J. [2001]. *The elements of statistical learning*, Springer Verlag.
- [9] Haykin, S. [2001]. *Redes Neurais: Principios e Prática. 2. Ed.*, Porto Alegre: Bookman.
- [10] Haykin, S., Thomson, D. & Reed, J. [2009]. Spectrum sensing for cognitive radio, *Proceedings of the IEEE* 97: 849–877.
- [11] Kadambe, S. & Jiang, Q. [2004]. Classification of modulation of signals of interest, *Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th* pp. 226–230.
- [12] Klautau, A., Jevtić, N. & Orlitsky, A. [2003]. On nearest-neighbor ECOC with application to all-pairs multiclass SVM, *J. Machine Learning Research* 4: 1–15.
- [13] Lynn, T. J. & Sha’amerr, A. [2007]. Automatic analysis and classification of digital modulation signals using spectrogram time frequency analysis, *International Symposium on Communications and Information Technologies, 2007. ISCIT '07.* pp. 916–920.
- [14] Mitola, J. & Maguire, G. Q. [1999]. Cognitive radio: making software radios more personal, *Personal Communications, IEEE* 6(4): 13–18.
URL: <http://dx.doi.org/10.1109/98.788210>
- [15] Muller, F., Cardoso, C. & Klautau, A. [2011]. A front end for discriminative learning in automatic modulation classification, *Communications Letters, IEEE* 15(4): 443–445.
- [16] Popoola, J. & van Olst, R. [2011]. A novel modulation-sensing method, *Vehicular Technology Magazine, IEEE* 6(3): 60–69.
- [17] Quinlan, J. [1993]. *C4.5: Programs for Machine Learning*, Morgan Kaufmann.
- [18] Ramkumar, B. [2009]. Automatic modulation classification for cognitive radios using cyclic feature detection, *Circuits and Systems Magazine, IEEE* 9(2): 27–45.

- [19] Shen, L., Li, S., Song, C. & Chen, F. [2006]. Automatic modulation classification of mpsk signals using high order cumulants, *8th International Conference on Signal Processing, 2006* 01.
- [20] Si, L.-L. M. X.-J. [2007]. An improved algorithm of modulation classification for digital communication signals based on wavelet transform, *IEEE Transactions on Aerospace and Electronic Systems* 03: 1226–1231.
- [21] Vapnik, V. [1995]. *The nature of statistical learning theory*, Springer Verlag.
- [22] Weka [n.d.]. <http://www.cs.waikato.ac.nz/ml/weka>.
- [23] Witten & Frank, E. [2005]. *Data mining: practical machine learning tools and techniques with java implementations*, Morgan Kaufmann.
- [24] Xiaorong;, H. T. J. [2004]. Modulation classification using arbf networks, *7th International Conference on Signal Processing, ICSP 04*. 03: 1809 – 1812.