

Characterization of Primary Users in Cognitive Radio Wireless Networks using Support Vector Machine

Danilo Lopez Sarmiento^{1*}, Edwin Rivas Trujillo² and Oscar Eduardo Gualdrón³

¹Department of Electronic Engineering, University Distrital Francisco Jose de Caldas, Bogota, Colombia, South America; dalopez@sudistrital.edu.co

²Department of System Engineering, University Distrital Francisco Jose de Caldas, Bogota, Colombia, South America; erivas@sudistrital.edu.co

³Department of System Engineering, University Pamplona, Pamplona, Colombia, South America; oscar.gualdrón@unipamplona.edu.co

Abstract

Objectives: This paper aims to solve a problem of existing research for Cognitive Radio networks Unlicensed (specifically for Wi-Fi with centralized network topology), which aims at the spectral making decision stage to efficiently characterize the behavior of users (called PUs) in order to identify spectral opportunities that can be used beneficially by other wireless applications or users. **Methods/ Statistical Analysis:** The methodology includes capturing the dynamics of channel usage by a user in the 2.4 GHz band, later to apply a pre-processing of the data and reach an estimate for the level of modeling (based training system) and prediction. **Findings:** The possible solution that was raised during the development of the research was to use a Support Vector Machine (SVM) to estimate the prediction by classification. **Application/Improvements:** The results suggest that the use of SVM does not correspond to the methodology of artificial intelligence accurately because the level of delivered estimate is not accurate.

Keywords: Characterization, Cognitive Radio, Prediction, Primary Users, Spectral Decision, Support Vector Machine, Wi-Fi

1. Introduction

This paper discusses the motivations and results obtained by using two classifiers based on SVM for making characterization (modeling and prediction) of PUs in Wi-Fi networks from the Emission/non-Emission of data on channel 6 (2.437 GHz Band). Figure 1 shows the application and research activities undertaken to characterize the use or spectrum.

Initially, a temporary series from emission data of a Wi-Fi network is generated. To this end, Acrylic¹ program is used, which returns a list of data indicating the date, hour, minute and second of emission or use of the channel by the PU (henceforth known as timestamp) and its length. With this data, a module has been implemented to load the file generated with Acrylic¹. Then another module was created to convert the information into a time series representing the emission or non-emission

for each time stamp (item 2). This time series is assumed to result in a real environment of a Cognitive Radio monitored and sensed. In regard to that time series representing the emission values (presence/absence) of PU, for characterization it has been integrated two SVMs, one internally generated from MATLAB (SVM MATLAB, now called the paper SVM-1)² (point 3) and other open source (LibSVM, now called SVM-2)³ (item 7). First it has been integrated MATLAB SVM (point 3), where the behavior is evaluated to estimate the channel usage varying multiple characteristics of the system: the number of iterations (point 4), the kernel type (item 5), conditions Karush-Kuhn-Tucker (KKT) (item 6), the characteristics of the data captured through the use of PCA (Principal Component Analysis) (item 9), change units of measure of the time series (point 10) and the length and number of examples (point 11).

* Author for correspondence

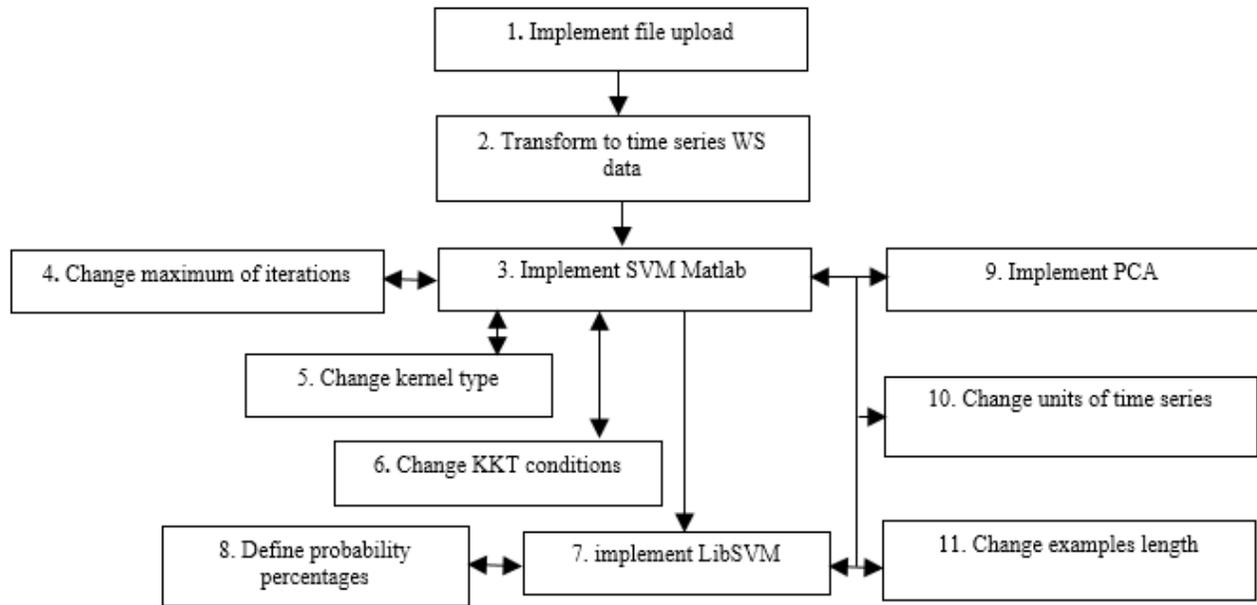


Figure 1. Block diagram of characterization system with SVMs.

Subsequently, LibSVM is used (item 7 in Figure 1) evaluating their level of predictability through the use of PCA (item 9) and varying: The change of units as of the time series (item 10), length and number of examples (point 11), it has also implemented the ability to alter the percentages of probability (item 8); while the latter parameter intends for the predictor return values close to 1, when the last moment of monitored time is emission and close to 0, when the last time point monitored is non-emission, regardless of the remoteness of the moment to predict the actual signal .

2. Characterization of Primary Users in Cognitive Radio

Currently, wireless networks and applications in much of the world have been characterized by a policy of fixed allocation of radio frequency spectrum regulated by the state. This fixed assignment causes the frequencies allocated for specific services are virtually obsolete and cannot be used by non-licensed users, even if they do not generate any interference⁴. According to studies by the Federal Communications Commission of the United States⁵ it has shown that much of the radio frequency spectrum is being used inefficiently⁶. Based on the temporal and geographical variations assigned spectrum utilization is about 15% to 85%, with a strong dependence

of time and space^{5,7}. Even more current measurements show that over 70% of the spectrum is not being used⁸. This inefficient and sporadic use of spectrum, along with the increase in demand for spectrum, has caused degradation for the quality of service in various networks and wireless applications, such as cellular network and Wi-Fi. This has motivated the development of recent research that found in the dynamic spectrum access the solution to the problem. The key technology that allows realizing dynamic access techniques to cognitive is radio spectrum⁴. The concept of Cognitive Radio (CR) was created in 1999 and is composed by the sensing steps, decision making, sharing and spectral mobility⁹. Decision making selects the best bandwidth available to Secondary Users (SUs) that use it opportunistically in the absence of PU; however, this task will depend a lot on how good the model characterization of PU is (i.e. how successful the modeling of PU dynamics and estimated future behavior). If the prediction is not as good, then the “best channel” amount of decisions will be very low and perhaps that efficiency is to be applied at that stage of the CR is then inefficient. The most representative methodologies studying the dynamics of PUs have focused their efforts on the use of techniques based on queuing theory, Poisson processes, stochastic models and static Bayesian networks^{9,10}; nonetheless it is essential to deepen in approach and application of methods grounded in artificial intelligence (with self-learning capacity) that

really allow to turn intelligent CR, thereby achieving that the system can autonomously adapt to changes in their environmental methods.

3. Proposed Algorithms as Classifiers in Primary Users

3.1 Previous Considerations of SVM as Classifier in CR

The Support Vector Machines are supervised learning algorithms that require training (modeling phase of PUs) with a set of examples before that can be applied to classify samples (phase prediction PUs). Each example has n well defined characteristics (in case of PUs corresponds to the emission values at the time before the prediction) and a value that defines the class, in this binary case (1 if the PU emit or 0 if not emit). The class is predicted from the characteristics. The characteristics are usually different measurements or values that define the example.

In the learning phase (modeling) characteristics are introduced (previous values of presence or absence of PUs) and class (the current state of issuance or non-issuance of PU) and SVM seeks a solution to differentiate examples by class. In the test phase, (check of accuracy rate of prediction) characteristics are introduced (previous emission and non-emission of PUs) of different examples in already SVM trained, and the classifier returns a class (1 or 0). For each example, the returned class is compared to the actual data to check the accuracy rate. Once satisfactory test results are achieved SVM can be used to classify examples (predict PUs). The SVMs have the main property of creating a hyper plane or set of hyper planes of greater dimensionality to offer input examples. Thus, the possibility of separation of classes increases dramatically. The SVMs are responsible to seek the greater distance hyper plane with examples of one kind and another. The simplest way to perform the separation is by a straight line, a straight plane or an N-dimensional hyper plane. Sometimes a SVM algorithm must deal with more than

two variables, nonlinear separation curves, cases where data sets cannot be completely separated into more than two categories. The Kernel functions offer a solution to this problem, projecting the information to a space with larger features. The Kernel function specifies how these spaces of higher dimensionality are created from the original dimensions.

Normally SVMs are used for regression and classification problems, but rarely used for time series prediction. To apply SVM characterization of cognitive radios, which are shown as a time series, a time series of n timestamps extracted each representing emission (1) or emission (0). The first $n-1$ timestamps are used as illustration features and timestamp n (the class) as the predicted value.

3.2 Sequence Diagram of SVM-1

The first algorithm used is supported on the existing own libraries in MATLAB². Figure 2 presents the flow diagram of how to implement and apply it to predict future patterns of behavior in PUs. Specifically, first a set of examples (point 1) is defined, representing the time series defining emission values (presence) and no emission (absence) of a PU for a range of specific time and past. All these examples characterize the PU, using the format required by SVM; i.e. extracting a time series of n time stamps, each representing a 1 or a 0. Then the kernel (item 2) function is defined. In case of a linear kernel, k is the scalar product. The SVM module uses an optimization method to identify support vectors, the weights a_i and bias b . These values do not represent anything of their own in cognitive radios, but are eigen values of SVM classifier, which help to generate two distinguishable sets of data, one for emission and one for non-emission. This optimization is repeated if valid values are not found up to the maximum iterations defined externally to the software application developed. This number of iterations is defined in paragraph 3.

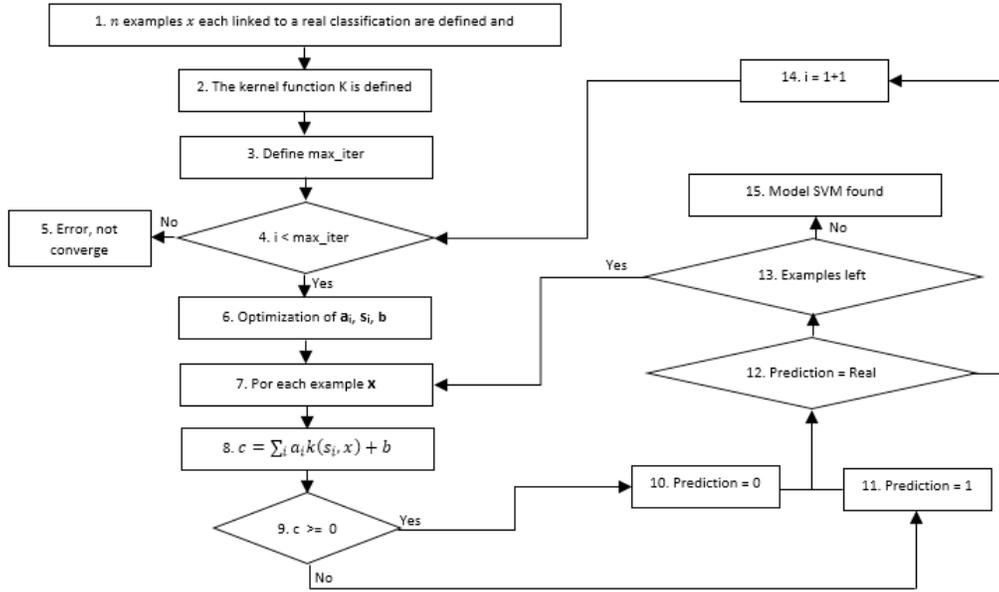


Figure 2. Flowchart of first SVM algorithm⁽²⁾.

For each iteration (point 4) it is done:

- Calculate the support vectors', weights a_i and bias b .
 - For each example x of those created in point 1 (point 7):
 - The prediction is calculated using the equation:
- $$c = \sum_i a_i k(s_i, x) + b \tag{1}$$
- The kind of prediction is obtained by analyzing the value c (item 9). If c is positive or zero, the class is 0 (item 10) and if c is negative, the prediction is 1 (item 11).
 - The algorithm checks the prediction with the actual classification of the example (item 12). If match is passed to the following example (item 13); if classes do not coincide, adds one to the number of iterations (item 14) and returns to point 4.

Should not be examples, or in other words, if all are classified correctly, the module returns an SVM model (item 15). If the maximum internal iterations of the SVM module are exceeded, it returns an error indicating that the algorithm has not converged, i.e. it has not found a suitable classifier for this problem (item 5). To characterize cognitive radios, x is defined as the characteristics of each example. Values, a_i, y are sought, so that a negative value c for all the examples of class 1 and a positive value c for

all examples of class 0 are obtained. Subsequently, with s_p, a_i, y, b defined, by applying the foregoing function to an example, the value c defines if the PU is in the channel or not. If c is positive or zero, the prediction is not emitted and if c is negative, the prediction is emitted.

3.3 Sequence Diagram of SVM-2

The second algorithm used and modified to model and predict the use of the channel is based on³. In Figure 3, the flowchart of the algorithm used is shown, which shows that the first set of n examples (representing time series defining the emission and non-emission for an interval of time) is defined. Thus, each of these samples is interpreted as absence/no absence of PU in an interval of time spent in the frequency band. Each time series consists of l emission values (1 if emissions; -1 if not) which must correspond with each x_i . The class of this example is represented by y_i which is the value of emission or no-emission at the time instant $l + 1$ (1). It is important to clarify that $x_i \in R^n, i = 1, \dots, l$; i.e. in the case of PU $x_i \in \{1, -1\}$ (item 2). $y_i \in \{1, -1\}, i = 1, \dots, l$ (item 3). In item 4, the type of kernel that will use the algorithm LibSVM is declared. Each kernel sets the way they are separating the sets by class y . The options are linear, polynomial, sigmoid and radial basis function. A linear kernel, which is given by the scalar product, since it was concluded (after testing) that was the best response delivered in modeling PUs and because it was lower computational cost which had been chosen. Mathematically, SVM-2 attempts to

solve the following optimization problem to characterize the PUs (point 5):

$$\min_{w,b,\varepsilon} \frac{1}{2} w^T w + c \sum_{i=1}^l \varepsilon_i \quad (2)$$

subject to: $y_i (w^T k(x_i) + b) \geq 1 - \varepsilon_i$

Where w is a matrix that defines the support vectors; c is the regularization parameter; $k(x_i)$ is the kernel function and b is the bias. These variables generate two differentiable data sets, one for emission and one for non-emission. As the matrix w can have a high dimensionality, before solving this equation is simplified by a mathematical transformation¹¹ applied to simplify optimization problems of high dimensionality. Simplification is in Equation 2 (item 6) of Figure 2.

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \quad (3)$$

Subject to: $y^T \alpha = 0, 0 \leq \alpha \leq C$

Where e is a vector of length n of “ones” and Q is a matrix $l \times l$ such that $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) \equiv k(x_i)^T k(x_j)$ (item 7). Q_{ij} Represents the ratio of similarity between two examples given class or predictive value and $K(x_i, x_j)$ represents the ratio of similarity between two examples. The conditions are

met to calculate α , the weight matrix, which defines the importance of each of the features.

Equation 4 calculates the support vector to obtain the optimum value of w :

$$w = \sum_{i=1}^l y_i \alpha_i k(x_i) \quad (4)$$

where $k(x_i)$ is the kernel applied on x_i characteristics of the examples, y_i is the real class of the example (1 or -1) and α_i is the weight vector calculated in item 6.

Finally, the decision function to determine whether an example will issue in the future is shown in Equation 5 (point).

$$\text{sgn}(w^T k(x) + b) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i K(x_i, x) + b\right) \quad (5)$$

It should be noted again that in characterization of PUs x_i is defined as the characteristics of each example (allowing only the values 1 or -1) and y_i as class, being 1 for emission and -1 as non-emission. Equation 5 calculates based on the entered data, if PU bears more similarities to the examples that emit in the future or which do not emit, thus predicts whether or not the presence of PU. In short, to characterize PUs the values w (support vectors), α_i (weight vector) and b (bias) calculated in the process described are used; subsequently the decision function is used to predict the channel use or not i .

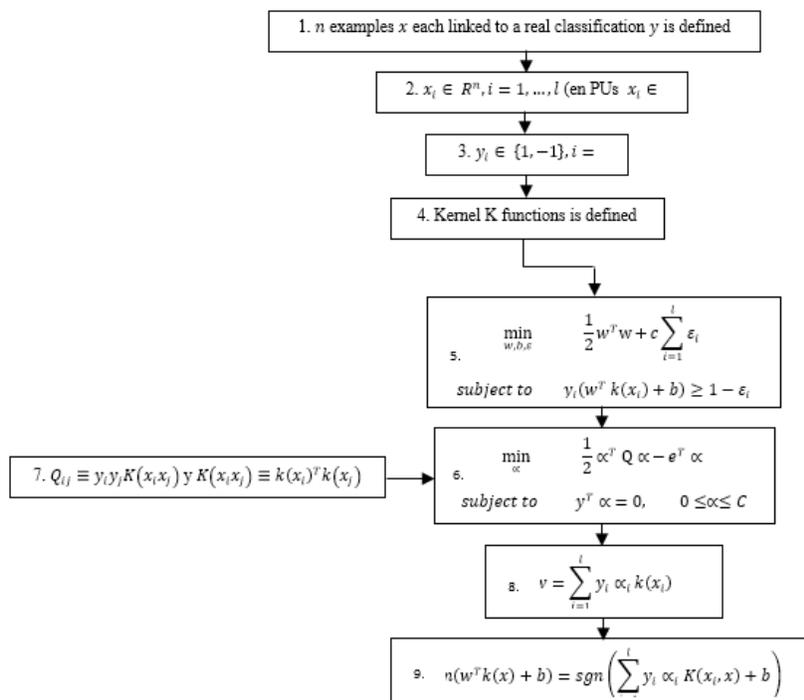


Figure 3. Sequence diagram of the second SVM proposed PUs⁽³⁾.

4. Testing Methodology SVMs

The methodology used to assess and analyze the level of characterization (modeling + prediction) of each SVM was based on the development of a software application on Matlab which included the 4 main modules shown in Figure 4.

- Load Acrylic file. This module loads a file in csv format Acrylic and extracts for each package its start time and duration.
- Transform time series.
- Modeling and estimation of PUs with SVM. Include: Generating the training cases; train the SVM and create test cases; predict the time series; return the actual test cases, prediction and success rate.
- Print Graphic. Where the two-dimensional space represents the axis of the independent variable time in milli seconds (msec) prediction, with 0 being the first estimate instantly. The more increases of value x , the farthest the prediction.

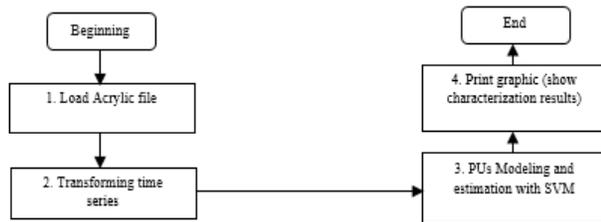


Figure 4. Methodology to estimate the absence or presence of PUs.

4.1 Preprocessing of Data

Once spectral occupation data is captured, it is proceeded to perform preprocessing thereof, with the aim that they were correctly interpreted by SVMs. This stage included points 1 and 2 of Figure 4. In Figure 5, the flow diagram module responsible for extracting data of Acrylic generated file and convert it to milliseconds (point 1) as

represented.

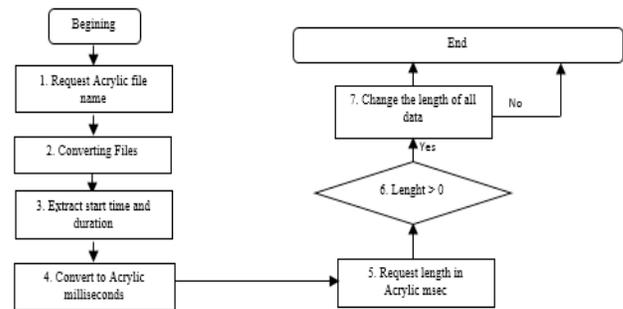


Figure 5. Diagram responsible for extracting the file data captured in Acrylic.

Specifically point 2 of Figure 5 is an action that calls the external library csv import². This library can read csv files and take them to MATLAB format; however, it is noteworthy that there has been a change (which eliminates all quotation marks extracted of csv to allow compatibility with files output protocol analyzer for different versions of Acrylic) for it to conform to the expected functionality.

Phases 5-7, give the possibility to choose a fixed duration in milliseconds of all PU states extracted. It can be entered at any value between 0 and 500. If the value is 0, no catches durations are changed, leaving its value as it appears in the original file. If a value between 1 and 500 is selected, the length of all packets will change to the defined value. For wrong values a value between 0 and 500 msec will be requested again. 1 ms as minimum value has been defined because it is the least because it is the lowest value that allows the data system used, and 500 milliseconds maximum value in order to limit the time processing required to perform training SVM.

Once data is collected, catches Figure 6 obtained are transformed by Acrylic to a valid format that allows modeling and predict through SVMs a time series of emission / non-emission of primary user¹² in the frequency band.

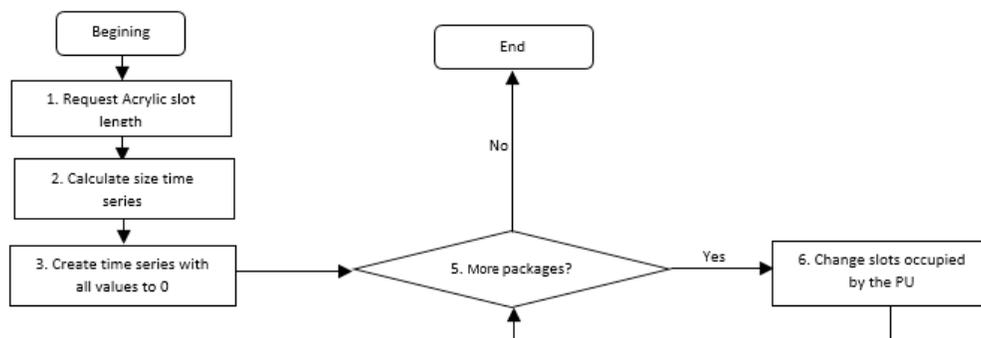


Figure 6. Processing time series sequences.

4.2 Estimation of PUs using SVM

Modeling and prediction of PUs in the Wi-Fi network include the sequence of steps specified in the block diagram of Figure 7.

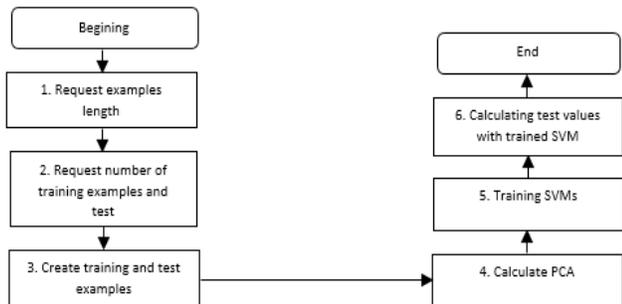


Figure 7. Sequence of steps to model and predict PUs with SVMs.

Initially, it is necessary to define the length of slots for training examples (modeling) and testing (prediction) of SVMs (point 1 in Figure 7). It was decided that this value must be a number between 10 and 2000, because the ideal time when there are changes between emission and non-emission is 1 second. Regardless, the chosen second in a time series extracted with the analyzer, there is usually a high percentage of emission and no-emission time; however, if smaller time such as 100 milliseconds is taken, there is a very high percentage that the chosen fragment is only emission or not emission only, which does not provide relevant information. Then the number of examples of training should be chosen (samples to model PU) and test (number of predictions made) whose maximum value was 1000, and to raise the value will imply extremely high performance times of more than three days (item 2). From item 3, many examples as desired from a defined length of the time series Acrylic captured are drawn. At point 4, the calculation of PCA can reduce

the dimensionality of a data set, by finding characteristics that affect its variability, through the construction of a linear transformation which includes the construction of a covariance matrix, the eigenvectors and Eigen values extraction as detailed in Figure 8. Dimensionality reduction applied to the examples of time series of the PUs having the same information in less amount of data, which could accelerate both processes characterization and prediction of PUs using SVM. The stages 5 and 6 (Figure 7) correspond to modeling the historical use of the channel (SVM training) and the calculation of prediction or estimation of future use of the spectrum band by licensed user.

The variables defining the blocks of the previous figure are: X is a matrix, X_i, j the j characteristic of example i and n the number of examples; μ_a is the Eigen value associated with P_a ; T_a , matrix $n * l$ representing the projections of X in μ_a . Figure 9 provides an overview of the flowchart to create the examples that are modeled and predicted. It can be synthesized according to the number and length of examples (presence/absence of PU), it is checked whether it may use the concept of sliding window for selecting the temporary subset. If $l + (1 * (n - 1)) > L$ can use sliding window, where l is length of the examples, n number of examples and L the length of the time series. In case of not being able to use the random method is used automatically to generate examples. If can be used by the sliding window, and a method of selection to start the temporary subset for each instance must be chosen. In case sliding window has been selected, it shall be calculated possible offsets that can be used depending on the length of the examples, the number of examples and the length of the time series, specifying an offset between 1 and the maximum calculated by the application.

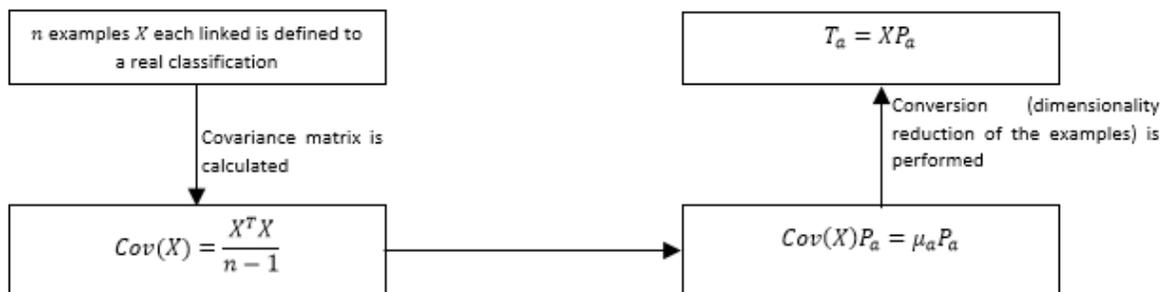


Figure 8. Data volume compression to be processed by the training stage and channel estimation ⁽¹³⁾.

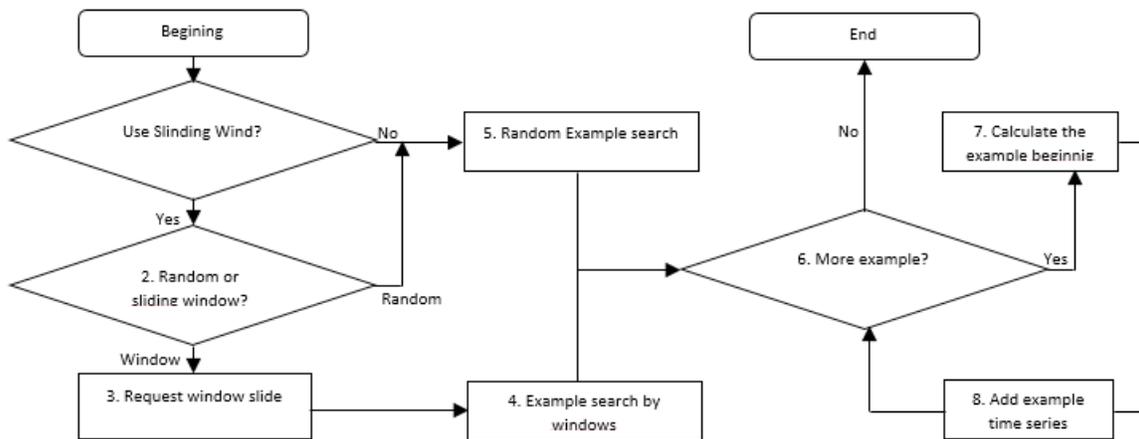


Figure 9. How to create examples to model and estimate the channel use of a primary user.

5. Evaluation and Analysis of Prediction Level Results with Algorithms SVM-1 and SVM-2

As mentioned above, the Acrylic data evaluation in principle transforms a time series from the MATLAB application developed in¹⁴ using the unit time as msec. Each packet lasts to be transmitted by many msec as their length in bytes or a fixed number of msec. As an example display in Figure 10, a time series is extracted from the file using Acrylic as packet transmission time length. Blue moments in which the PU is present at the channel are observed, while the white areas show times no emission or waste of bandwidth is available on the channel. The total length of the time series is 600,296 units of time.

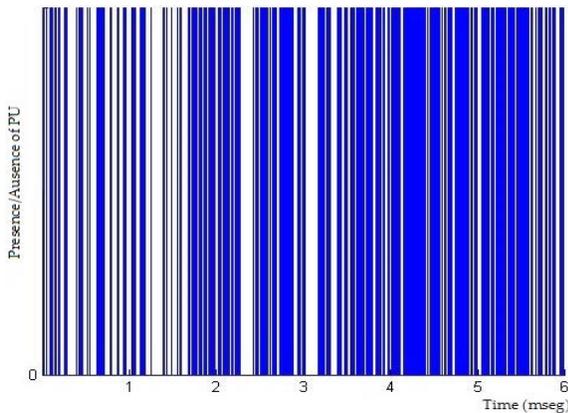


Figure 10. Behavior of a PU in terms of emission/non-emission for a channel on the Wi-Fi band delivered by Acrylic.

In Figure 11, a fragment of the time series expanded from 10,000 to 40,000 milliseconds is observed. As a representative characteristic that there are phases shipping and no shipping of long and short data is observed.

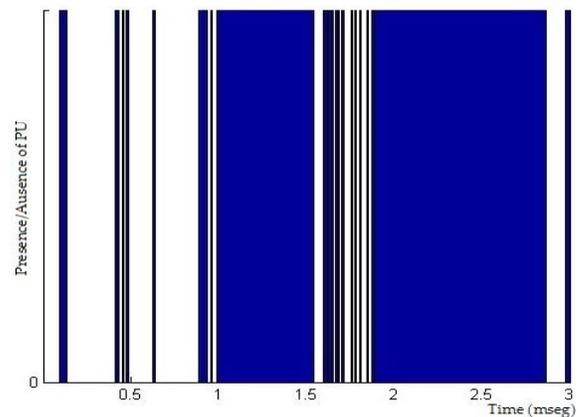


Figure 11. Behavior emission/non-emission of PU between 0 and 3 msec.

5.1 SVM_1

Trained for the SVM (section that attempts to model the PU) leads to the conclusion that it does not converge. If SVM does not converge, means it fails to find any combination of support s_i , weights a_i and bias b vectors that meet the specified cost and thus find a pattern that accurately represents the PU. In order to improve the level of PUs modeling various solutions tested:

- Modify the length of the examples (10 to 2000) and the number of examples (of 1000-100000) providing more information about the modus operandi of PU.

- The maximum number of training iterations is increased to a maximum value (15,000 to over two billion), which triggers the execution time of the algorithm.
- The kernel of the algorithm is changed, from linear (which seeks a linear hyperplane to separate the two sets of examples) to polynomial of three degree at hyperplane.
- Karush-Kuhn-Tucker conditions (KKT)¹⁵ are transformed, which are necessary and sufficient conditions for the optimal solution of a mathematical programming problem (characterization of PUs). In the case of SVM implemented in Matlab, these conditions are applied to the Lagrangian (Equation 6) to calculate the maximum separation hyperplane.

$$L(x, \lambda) = f(x) + \sum \lambda_{g,i} g_i(x) + \sum \lambda_{h,i} h_i(x) \quad (6)$$

Where $f(x)$ the kernel to optimize, $g(x)$ a vector of restrictions on the type $g(x) \leq 0$, $h(x)$ a vector of restrictions on the type $h(x) = 0$. The λ vector, correspond to the Lagrange multipliers. Note that the KKT conditions used by the SVM are:

$$\begin{aligned} \nabla_x L(x, \lambda) &= 0 \\ \lambda_{g,i} g_i(x) &= 0 \forall i \\ g(x) &\leq 0 \\ h(x) &= 0 \\ \lambda_{g,i} &\geq 0 \end{aligned} \quad (7)$$

By modifying the margin of error of KKT conditions, it is allowing a margin of error in the location of possible existing pattern in the behavior of PU, which means that modeling is less conformed to the actual data.

- The rate of violation of the conditions KKT, which specifies the fraction of the number of variables that cannot meet is changed. 0 implies that they must fulfill all the conditions and 1, which cannot be met by any.
- To find the hyperplane, SVM allows two approaches: the first is Sequential Minimal Optimization (SMO), which bases its operation on the Lagrange multipliers, and the second Quadratic Programming (QP), which is an optimization system included in the Optimization Toolbox license).

Including the above modifications and trained system, the SVM converge is not achieved, or in other words, which can successfully characterize the PU. That is why this SVM is not possible to predict the PU.

5.2 SVM-2

While tested to perform modeling of the PU, then to make a prediction with the maximum number of examples (100,000) and the maximum length (2,000) according to each application. The duration of training was 2956 seconds, obtaining a satisfactory result. Subsequently, a calculation for 1000 examples for the prediction at time $l + 1$ was experienced (corresponding to an estimate short range); the result of success is 100%:

```
Predicción SVM 1 de 1
Accuracy = 100% (1000/1000) (classificaton)
Elapsed time is 12.364821 seconds.
>>
```

In Figure 12, the above prediction example is graphically located (assuming that the modeling had a 100% success). It should be mentioned that as the predicted time series consists of a single value, instead of a line appearing, a single point appears. The blue sign shows the actual time series of data sending, as to be the same value as the prediction, appears not shown, for the prediction in red is exactly about the real value. The axis of the independent variable, represents the time in prediction milliseconds, with 1 being the first and only prediction time (instant $l + 1$). The dependent variable, identifies the shipping values and no shipping of PU, being 1 the existence of PU and 0 non existence. In the classification matrix (Table 1), the classification results for a thousand examples of prediction ($l + 1$) are compiled; the rows of the matrix represent the values of model estimation, whereas the columns indicate the actual values. The classification matrix is created by sorting all cases into categories: whether the predicted value matches the actual value, and if the value of prediction is right or wrong. In this case, in 487 examples there was no real issue, and they have all been predicted as emission, while in 513 examples there is no real emission, and classifier was classified as non-emission. Not having misclassified either case, the prediction has a 100% success.

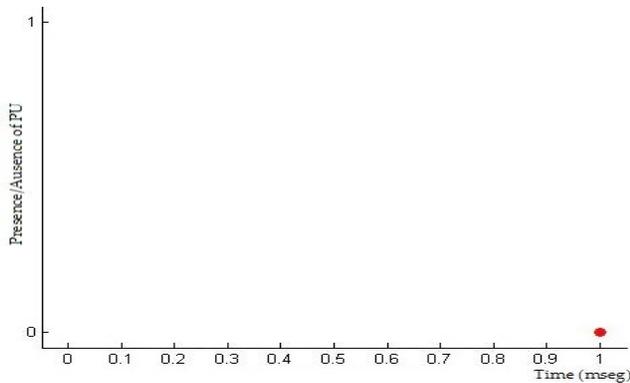


Figure 12. Comparing the actual behavior and the prediction for the PU in the evaluated channel without the use of PCA.

Table 1. Classification matrix for 1000 examples

Predicted/Real	1 (real emission)	0 (not real emission)
1 (predicted emission)	487	0
0 (not predicted emission)	0	513

It includes the previous PCA process seeking to reduce the training time of SVM system. PCA components calculate the maximum number of examples (100,000) and the maximum length (2000) takes 211 seconds. It should be kept in mind that before generating the modeling, the mean value is removed, and their respective components PCA is calculated. This process takes approximately 65 seconds to 100,000 examples having a length of 2,000 units. The training duration is 3958 seconds or 66 seconds. Tested to calculate 1000 examples evaluated at time $l + 1$. The result of success is 100%:

```
Predicción SVM 1 de 1
Accuracy = 100% (1000/1000) (classificaton)
Elapsed time is 11.493758 seconds.
>>
```

The classification matrix (Table 2) indicates that in 356 examples there is actual emission and all have been predicted as existence of the PU in the channel, while there is no real emission and the classifier has cataloged 644 emissions as examples. Not having misclassified either case, the prediction has a success of 100%.

Table 2. Classification matrix for 1000 examples using PCA

Predicted/Real	1 (real emission)	0 (not real emission)
1 (predicted emission)	356	0
0 (not predicted emission)	0	644

In concluding that the percentage is 100% successful, it is attempted to detect the time instants $l + 1000$. For this first time $l + 1$ is calculated, then this prediction is used to calculate the time $l + 2$ and so on, up to $l + x$. The training process or characterization is exactly as in the previous tests. For $x = 1000$ the result of the prediction is about 50% success without PCA and PCA (Tables 3 and 4).

Table 3. Classification matrix for a thousand examples with estimation $l + 1000$ and without using PCA

Predicted\Real	1 (real emission)	0 (not real emission)
1 (predicted emission)	289	263
0 (not predicted emission)	229	219

In 518 examples there is real presence of PU; 289 of them have been correctly predicted as emission and 229 of them have been classified incorrectly as non-emission. In 482 examples there is no real emission and classifier has classified 263 wrongly as emission and non-emission correctly as 219. 492 cases have been classified incorrectly and 508 cases properly, so in this case we have 50.8% accuracy:

```
Predicción SVM 1000 de 1000
Accuracy = 50.8% (508/1000) (classificaton)
Elapsed time is 15.095385 seconds.
>> |
```

Table 4. Classification matrix for a thousand examples with estimation $l + 1000$ using PCA

Predicted\Real	1 (real emission)	0 (not real emission)
1 (predicted emission)	270	242
0 (not predicted emission)	243	245

In 513 examples there is real emission. 270 of them have been correctly predicted as emission and 243 of them have been classified incorrectly as non-emission. In 487 instances there is no real emission and classifier 242 wrongly classified as emission and non-emission correctly as 245. 485 cases have been classified incorrectly and 515 cases properly, so in this case we have accuracy of 51.5%:

```
Predicción SVM 1000 de 1000
Accuracy = 51.5% (515/1000) (classificaton)
Elapsed time is 16.175938 seconds.
>>
```

For each trained and estimated cases, percent always close to 50% due the probability to be emitted or not. Basically the classifier characterizes the PU signal so that expects the signal is eternally maintained at 1 if started with 1, and maintained at 0 if started as 0, as evidenced in Figure 13, where the predicted signal (red) is kept in non-emission the entire prediction, while the actual (blue) not evidence as the presence of PU until after 700 millisecond, returned to the non-emission state from 965 milliseconds.

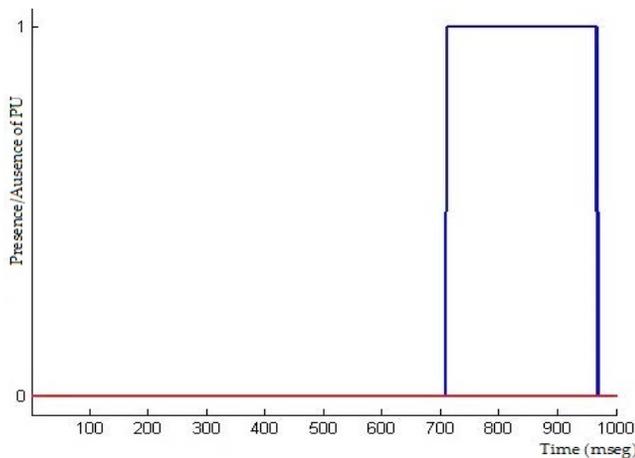


Figure 13. Comparing actual activity of PU and estimated by the SVM-2 (using PCA).

Finally tested in the prediction phase, extract percentage probability that each sample is in one class or another (i.e. the aim is to obtain a percentage probability in the emission, instead of indicating whether presence or absence, in this way instead of estimating a 1 or 0, a value x between 0 and 1, where the value x is the emission probability at the moment to predict and $1-x$ the probability of non-emission to be predicted). The algorithm to return percentages is trained. The process takes 12.7 hours in training without applying PCA and 10.5 hours using PCA (which makes the computational consumption SVM training is triggered). For a $x =$ prediction 1000 msec in the future, the result is about 50% success without PCA and 50% success with PCA. When percentages are used, there is a probability that the signal changes from 0 to 1 and vice versa, but the probability is around 0.15% and the possibility of returning to the original value in the next moment is very high, which is why in most cases the signal remains at its original value; This is evident in Figure 14 which shows that at predicting a certain time, tries to switch to a high level

(on the timestamp 850), but because of the previous history, it quickly rectifies and returns to the low level, confirming that trying to characterize the activity of PUs by the Support Vector Machine is inefficient.

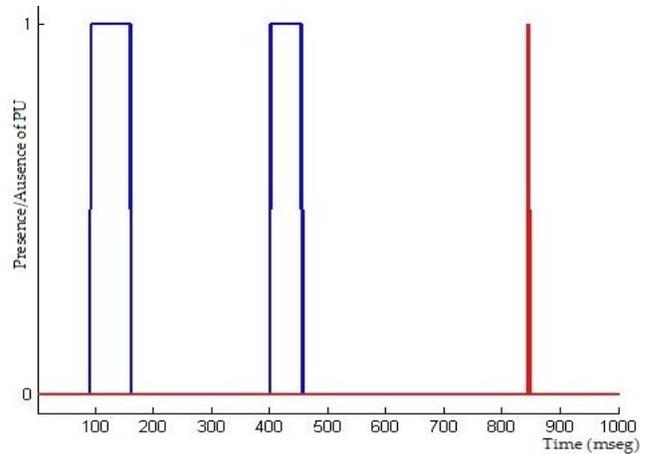


Figure 14. Comparison of the actual activity of PU and estimated by the SVM-2 based on percentages of probability.

6. Discussion

The models presented have pretended to create a new proposal for characterization (modeling + future estimate) of PUs for a channel in the spectral band Wi-Fi based on the use of supervised learning algorithm Support Vector Machine.

The results from the use of SVM-1 have failed to represent the dynamic behavior of PU since the algorithm has been unable to converge, thereby indicating that it cannot find a solution to the problem preventing entering the prediction phase.

A second algorithm (SVM-2) has been evaluated, which can reproduce the expected dynamic and is able to deliver reliable results for the first timestamp of future estimates, however, when estimated n time stamps consecutive for the same time series, it is concluded that the algorithm always predicts based on the last instant of time recorded (if it is present of PU, it always predicts their presence, if the channel is free, predicts absence of PU). From the approaches and analyzes presented in the article, and the experience gained over the investigation developed, it shows that the use of artificial intelligence techniques for modeling and estimation of primary users in cognitive networks (with network topologies centralized) could be a great success given the ability of autonomous learning they have, by using methodologies

such as dynamic Bayesian networks or neural networks¹⁶ (perceptron pyramidal multilayer (PMPG) and Long Short-Term Memory (LSTM), all in favor of continuing to generate proposals to optimize the management of usable spectrum for wireless communications by using cognitive radio.

7. Conclusions

After performing various and varied tests on SVM, it has not been able to reach a useful result with this system. The reason why SVM is not practical to characterize and predict PUs in cognitive radios is that it requires many examples and predictions are fixed, distributions are not taken into account (which is probably what best fits to describe a cognitive radio). However, although the prediction is fixed, the used SVMs allow to extract what the chances are that the example applied is class 1 or class 0; yet the percentages obtained were always values greater than 0.995 for one of the two classes, either emission or non-emission. That is why SVM-2 (known as LibSVM) is able to predict how the PU will behave, but somehow wrong, because if the system is emitting in the last moments recorded, the SVM detects as a prediction that radio will be emitting infinitely. The same happens if the PU to predict is non-emission at the end, for example, the prediction offered by the algorithm is that the PU will be without emission permanently.

Another reason why the SVM-based systems are not good predictors in the characterization stage within the phase spectral decision cognitive radios, may be that the features defined for each example are 0 and 1 in a time instant earlier, making all characteristics show homogeneous values; if there is a heterogeneity (which could describe a different aspect of the example environment) SVM would be easier to distinguish between classes and make a more accurate prediction. The above hypothesis, allows speculating that a broader spectrum of values helps to classify the SVM rate of success, and it would make more information available. Yet this problem can be remedied by introducing more features, but the environment in which it is performed has limited storage space.

Finally, it is assumed that SVM and any classifier analyzes an input dimension to specify which class defines these dimensions; however, in this case, what it has is a time series, which will not have classes as such, but an

accumulation of points that determine emission pattern, so it would be more interesting to consider the prediction as an adjustment to a pattern time series.

8. References

1. Wi-Fi scanner for 802.11 ac networks. Acrylic Wi-Fi home. Apr. 2016. Available from: Crossref
2. Mathworks (support documentation). 2012. Available from: Crossref
3. Chang CC, Lin JL. LIBSVM: A library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*. 2011; 2(3):1–27. Crossref
4. Akyildiz IF, Lee WY, Vuran MC, Mohanty S. A survey on spectrum management in Cognitive Radio networks. *IEEE Communications Magazine*. 2008; 46(4):40–8. Crossref
5. Federal Communications Commission. Notice of proposed rulemaking and order. Mexico D.F: Report ET Docket. 2016. Crossref
6. Hernandez C, Paez I, Giral D. AHP-VIKOR model for spectrum handoff in Cognitive Radio networks. *Revista Tecnura*. 2015; 19(45):29–39. Crossref
7. Granelli G, Sherman S, Mody M. IEEE Standard definitions and concepts for dynamic spectrum access: Terminology relating to emerging wireless networks, system functionality and spectrum management. *IEEE Standard*. 2010; 20(2):1–9.
8. Sahai A, Hoven N, Tandra R. Some fundamental limits on cognitive radio. Department Electrical Engineering and Computer Science. University of California. Apr. 2005; 1:464–9.
9. Lopez D, Trujillo E, Gualdrón O. Elementos fundamentales que componen la radio cognitiva y asignación de bandas espectrales. *Revista Informacion Tecnologica*. 2015; 26(1):23–40. Crossref
10. Saleem Y, Rehmani MH. Primary radio user activity models for Cognitive Radio networks: A survey. Elsevier. *Journal of Network and Computer Applications*. 2014; 43(1):1–16. Crossref
11. Meerschaert MM. *Mathematical modeling*. Academic press is an imprint of Elsevier; Fourth ed. 2013. p. 21–50.
12. Sharma H, Kumar K. Primary user emulation attack analysis on cognitive radio. *Indian Journal of Science and Technology*. 2016; 9(14):1–6. Crossref
13. PCA (Principal-component-analysis). 2016. Available from: Crossref
14. Mathworks. MATLAB mex support for visual studio (and mbuild). 2013. Available from: Crossref
15. NTNU, Department of mathematical sciences. 2015. Available from: Crossref
16. Xing X, Jing T, Cheng W, Huo Y, Cheng X. Spectrum prediction in Cognitive Radio networks. *IEEE in Wireless Communications*. 2013; 20(2):90–6. Crossref