Artif Intell Rev
https://doi.org/10.1007/s10462-017-9600-4

Primary user characterization for cognitive radio wireless networks using a neural system based on Deep Learning

Danilo López[1] · **Edwin Rivas**[1] · **Oscar Gualdron**[2]

Abstract Cognitive radio is a paradigm that proposes maximizing the utilization of the usable radio-electric spectrum, allowing licensed users (PUs) and non-licensed users (SUs) to simultaneously coexist through the dynamic management and assignment of spectrum resources, by integrating the stages of spectrum sensing, decision, sharing and mobility. Spectrum decision is one of the most important stages, but its optimal operation depends on the characterization sub-stage, which is in charge of efficiently estimating time gaps in which a PU won't make use of the assigned spectrum, so that it can be used in an opportunistic fashion by SUs. The design and implementation of an algorithm based on the Long Short-Term Memory (LSTM) recurrent neural network is proposed in order to increase the success percentage in the forecasting (presence/absence) of PUs in spectrum channels. The accuracy level exhibited in the results indicates LSTM increases the prediction percentage as compared to the Multilayer Perceptron Neural Network (MLPNN) and the Adaptative Neuro-Fuzzy Inference System (ANFIS) learning models, which means it could be implemented in cognitive networks with centralized physical topologies.

Keywords Cognitive radio · Long-Short Term Memory · Neural network · Deep Learning · GSM

✉ Danilo López
dalopezs@udistrital.edu.co; ingeniero24@hotmail.com

Edwin Rivas
erivas@udistrital.edu.co

Oscar Gualdron
oscar.gualdron@unipamplona.edu.co

[1] Faculty of Engineering, Distrital University, Bogotá, Colombia

[2] Faculty of Engineering, Pamplona University, Pamplona, Colombia

Published online: 11 December 2017

1 Introduction

Recent advances in the wireless communications field, such as Dynamic Spectrum Access (DSA), offer the promise of creating new methodologies for solving some of the main problems wireless access technologies face today. One of them has to do with the assignment of the portion of the usable radio-electric spectrum (licensed and unlicensed) that facilitates the connection of devices with low error probabilities. The distribution of these frequency ranges is currently controlled by government entities in each country, where each telecommunications operator is individually assigned a fixed range of frequencies through renewable licenses. This static assignment, coupled with the demand from new technologies and more robust applications, has triggered the demand for more spectrum to support the volume of traffic circulating on the networks, and generating an apparent spectrum shortage. Multiple studies have validated the theory that the real problem lies in the way the spectrum is distributed, with large portions that are under-utilized, and segments with use varying from 15 to 85% Shared (2015), Federal (2003), IEEE (2008), Sahai et al. (2015). Managing the spectrum resource more efficiently would be an important step for supporting the demand for bandwidth. In this regard, the opportunistic access to spectrum through Cognitive Radio Fortuna and Mohorcic (2009), Popescu et al. (2014), Masonta (2013) is a technique that is available to dynamically manage the spectrum and optimize its exploitation.

Dynamic spectrum assignment in Cognitive Radio (CR) is supported by the four stages described in Khalid and Anpalagan (2010), Akyildiz et al. (2009, 2014). According to pez et al. (2015), the spectrum decision stage, one of the stages on which less research has been done, is in charge of selecting and assigning available channels or frequency bands (not used by licensed users (PUs)) so that they can be used in an opportunistic fashion by non-licensed users (SUs) to transport information. Since there's no guarantee a channel will be available all the time a SU needs it, it's important to determine the pattern of future PU arrival in the spectrum. Using CR's learning ability, the historic information on the use of the resource can be used to predict its future profile Masonta (2013). This is achieved by PU characterization (modeling and prediction).

The base station of the CRN (Fig. 1) can, taking PU activity into account, decide on the best channel/channels for SUs. In this context, and according to the description in Fig. 1, success in the efficient selection of channels will depend on the reliability of the forecasting algorithm for predicting the presence/absence of licensed users.

If the prediction percentage is high, system operation will be optimal because the probability of erroneously assigning spectrum will be very low, and collisions between the PU and SUs will be avoided. On the other hand, if the estimation is not accurate, as occurs in multiple state of the art models that were reviewed, the system will not operate appropriately, and the amount of interferences produced will make the implementation of cognitive wireless networks unacceptable. Providing an adequate solution to the characterization problem involves the possibility of exploiting the autonomous learning skills of artificial intelligence (AI). In this regard, the following question arises: Does implementing the Long Short-Term Memory learning model as a predictor allow an improvement in the accuracy percentage when detecting the presence or absence of PUs in spectrum bands? The ability to learn (through the inclusion of peephole connections), store patterns during network training, and access information stored for long periods of time in LSTM memory cells are characteristics that can be applied to the chaotic behavior of PU signals, seeking to more accurately estimate the arrival pattern of PUs in the assigned spectrum bands.

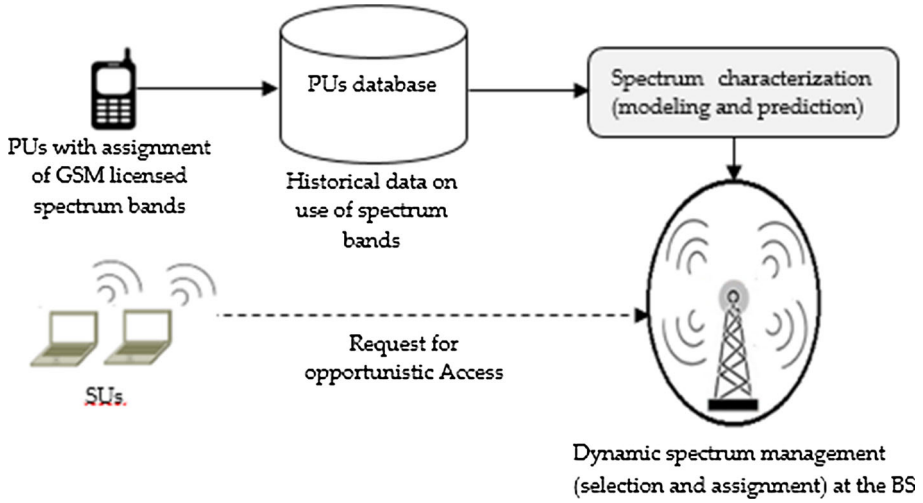

Fig. 1 Contextualization of characterization in the spectrum decision stage for CRNs

This paper seeks to develop and evaluate the performance of a LSTM-based algorithm that is capable of modeling and predicting with a greater accuracy (taking the state of the art as a reference) PU activity in cognitive radio wireless networks. The proposed solution to the problem is new and relevant because there is no evidence of the use of LSTM in CRNs, much less in the characterization of licensed users. Therefore, there is a potential for interest in the scientific community around the generation of Deep Learning based models that could increase the performance of cognitive networks in each one of the constituent stages.

From the preceding analysis, it's clear that, even though there are several proposals for modeling primary user activity, it's important to continue searching for a way to minimize the error percentage in PU prediction, which in turn will result in the optimization of the spectrum decision stage for CR. That is the focus of this research paper, which contains the following: Sect. 2: the state of the art related to PU characterization in CR; Sect. 3: the development of the proposal; Sect. 4: results evaluation and analysis; at the end, discussion and conclusions (Fig. 2).

2 Scientific review

The main research proposals for modeling primary user activity in licensed and non-licensed bands are found and explained in pez et al. (2015), López et al. (2016, 2017), Tumuluru et al. (2010), Pattanayak et al. (2013), Bkassiny et al. (2011), Khabazian et al. (2012), Banaei and Georghiades (2009), Gutiérrez et al. (2013), Wang et al. (2011), Yarkan and Arslan (2013), Uyanik et al. (2012), Ghosh et al. (2010), Lee and Akyildiz (2011), Canberk et al. (2011), and are supported by the methodologies shown in Fig. 3. They are detailed briefly in Sects. 2.1– 2.4.

2.1 Learning-based characterization algorithms

The authors of the article in López et al. (2016) developed an algorithm based on Bayesian networks (BN) for characterizing PUs in an infrastructure-based GSM-850 network. The

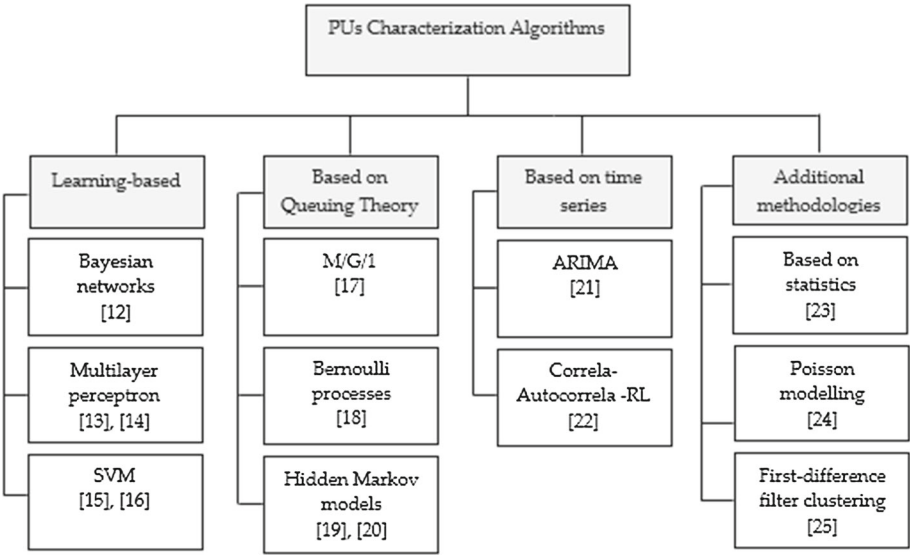

Fig. 2 Algorithms for PU characterization in spectrum bands

prediction calculation uses the following network variables for reference: delay, bandwidth, signal-to-noise ratio and channel occupation/non-occupation. The Tree Augmented Network (TAN), Expectation Maximization (EM) and Gradient Descendent (GD) methodologies were used to determine the accuracy level of BN training. The Confusion Matrix (CM) and Error Rate (ER) metrics and other metrics were used to establish the accuracy level of the prediction. The results show that the prediction achieved by the BN is quite good compared to the existing state of the art. This indicates that the BN, as a prediction model, may provide higher levels of PU characterization and CR performance. However, the authors wish to point out that it is necessary to test the algorithm with other types of data flows.

In Tumuluru et al. (2010), the characterization problem is described in terms of the lack of *a priori* knowledge of traffic characteristics, and a predictor is designed based on the neural network model of an unsupervised multilayer perceptron (MLP), where there will be a desired value and its estimation; the difference between the two will be the error. The training algorithm acts at that point with the purpose of minimizing the error rate as much as possible by creating a correspondence function between the input vector and the desired value. The parameters are updated repeatedly until reaching the minimum mean squared error or, otherwise, the maximum number of iterations; once the training is finalized, the predictor is tested by means of random observations of subsequent predictions. The conclusion reached through simulations is that the calculation of erroneous predictions can be improved if each interval of the MLP predictor has a short duration, spectrum utilization improvement is more than 60%, and the energy reduction percentage for traffic intensity detection is 50% less than the reduction observed without the use of the MLP predictor.

In Pattanayak et al. (2013), an artificial neural network (ANN) model is used to predict channel status in a television band; in order to meet the goal, parameters such as the channel spectrum efficiency and the distance (d) between the primary base station and the secondary station are taken into account. The proposed ANN model predicts channel status as "1" for a busy channel and "0" for an available channel. The exploration system analyzes the licensed

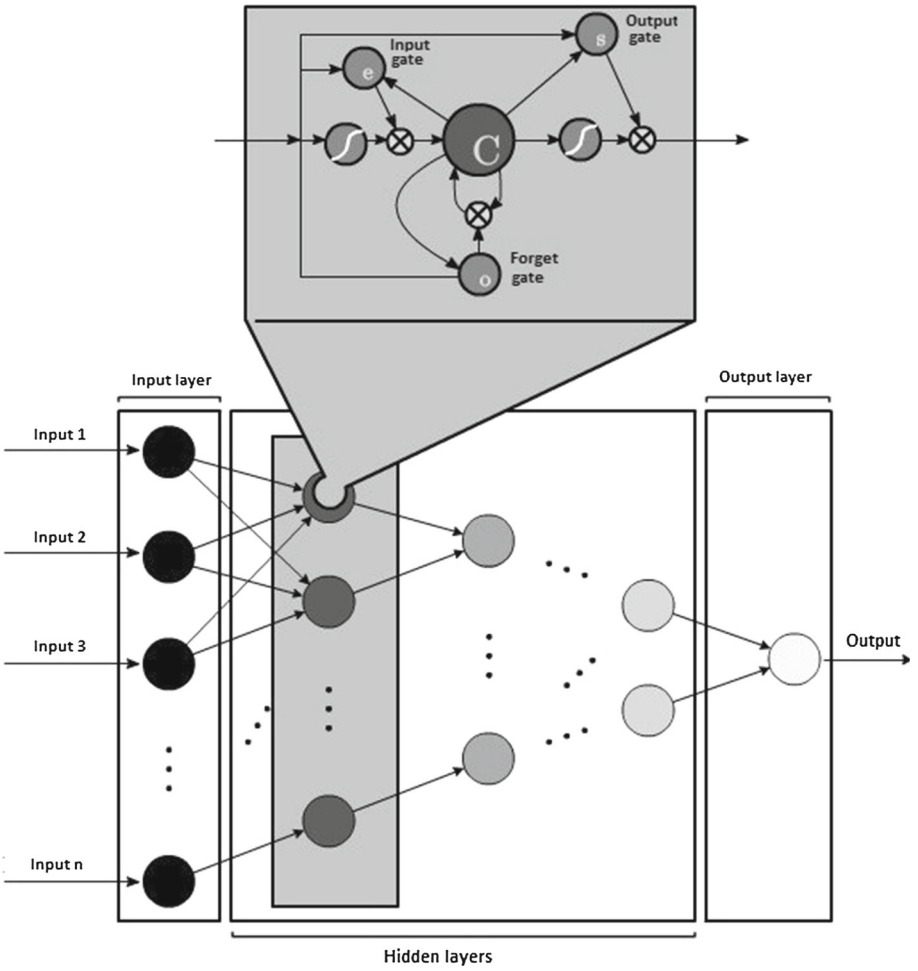

Fig. 3 Graphical representation of LSTM-type neural networks

channels, taking into account that network variables such as the signal noise ratio (SNR), the bandwidth efficiency, the channel capacity, and the distance are provided as inputs to the ANN. All the parameters that are provided as inputs are indices for identifying blank spaces. Thus, it is concluded that the designed ANN model can identify spectrum holes for an SNR of up to − 20 dB. The neural network model that is trained and optimized the first time does not need to be trained further, and the optimized weights can be continuously stored for subsequent predictions, making this specific technique the most appropriate for detecting the spectrum and assigning it in the primary television station's coverage area.

In the state of the art discussion in Bkassiny et al. (2011), the authors stress the importance of autonomous learning in CR, taking into account that said learning implies that current and future characterizations and actions are obtained from past observations and behaviors, without confusing this with the reasoning posture (consisting of observing the current state of the environment and making decisions without necessarily taking the past into account), which is why the development of learning is essential for optimizing processes in CR; this learning

may be supervised or unsupervised. Supervised learning is characterized by its use in known settings and the use of techniques such as: artificial neural networks (that can be used for spectrum modeling, characterization and decision at runtime, thus achieving a certain degree of learning), Support Vector Machine (SVM) (used as a signal classifier in CR settings); it's important to point out it's used by agents that learn autonomously without supervision, where said agents will base the learning process on their interactions with the environment using dynamic programming or time difference, or the trial-and-error method. On the other hand, unsupervised learning is useful if CR has no knowledge of certain environmental parameters and tries to learn an optimal action policy that leads to better system performance.

In the article in López et al. (2017), the authors socialize the results of implementing an SVM-based algorithm in the spectrum decision stage for a wireless CR with an infrastructure-based topology. The characterization sub stage of the system that was modeled includes the use of a database with real information on PU behavior, as well as the generation of potential secondary users (SUs) differentiated by service quality criteria requested from the base station. In the channel selection sub stage, an algorithm for predicting the arrival of SU requests at the base station was designed using a multilayer perceptron neural network. The authors conclude from the discussion that a SVM-based decision making system is optimal because the amount of collisions between PUs and SUs is less than with other techniques, although the training or learning time may be greater.

2.2 Queuing-theory-based characterization algorithms

An analytical study for a hybrid network based on the IEEE 802.11 standard is addressed in Khabazian et al. (2012). In order to preserve the priority of PUs, it was assumed that SUs compete for the opportunistic exploitation of channels when channels are free of any activity by licensed users. Each PU is modeled as a M/G/1 discrete queuing system with an arrival rate of λ packages/second and a service rate of $1/E\,[D]$, with $E\,[D]$ defined as the spectrum access delay for PUs in the presence of SUs. The value shows the mean of the time interval between the arrival time of a package to the main queue of a PU and the time the PU accesses the channel in order to carry out a transmission. According to the authors, the value of $E\,[D]$ associated with the queuing delay directly impacts the average package delay $E\,[D_t]$. Thus, it can be inferred that the performance of the primary network with a given package arrival rate may be affected depending on the package's useful load, and the number of neighboring SU nodes that exist on the cognitive network (as demonstrated by the results of simulations that were applied).

In cognitive systems, the MAC (Media Access Control) protocol is the standard. It has the ability to manage the interference level for PUs and coordinate access to the radio-electric spectrum for SUs. Therefore, the authors of Banaei and Georghiades (2009) propose the use of a MAC Overlay cognitive standard that allows finding and using available network spectrum for opportunistic exploitation by SUs. In order to reduce interference among secondary users, the authors use the persistent CSMA (Carrier Sense Multiple Access with Collision Detection) standard. The system model considers a frequency spectrum consisting of N independent channels with W_j, $j \in N$ bandwidths. The total available bandwidth for primary users is $W = \sum_{j \in N} W_j$, where each channel supports a single PU. The channel occupation statistics are independent from each other, and follow a two state Bernoulli distribution with probability P'_j, $j \in N$, with the characteristic that the PUs are distributed according to a two-dimensional Poisson process with density λ. The results show that SUs are perfectly capable of detecting primary signals within the nodes' operation radius (R), and accessing channels in the absence of PUs without creating interference.

Although most of the research on cognitive radio focuses on frequency bands above the upper limit of high frequency (HF) Gutiérrez et al. (2013), CR principles can also be applied to HF band communications in order to use the spectrum more appropriately based on regulatory and propagation restrictions. Reference Gutiérrez et al. (2013) considers that users inherited from other frequencies are PUs who transmit without resorting to any intelligent procedure. The HFDVL architecture (HF voice and data transport using 3 KHz bandwidths) is used for SUs. The goal of the study is to improve spectrum use efficiency by detecting the future presence of PUs in channels (to avoid collisions), while the information from SUs is transmitted through different channels using the transceiver. To that end, a dynamic algorithm that monitors the activity of the PUs Wang et al. (2011) was developed, and estimates short term future predictions for presence time using the Hidden Markov Model (HMM). The system is trained for real values obtained in the amateur radio band in the 14 MHz frequency in three different scenarios: available, partially available, and unavailable channels. The validation of results was based on predicting activity on a channel during the next minute, reaching an average prediction error equal to 10.3% when prior knowledge of the activity has a 1 min duration, and reducing the value to 5.8% when the prior time for analysis is 8 min.

2.3 Time-series-based characterization algorithms

The statistical approach based on binary time series that is described in Yarkan and Arslan (2013) provides the deterministic and non deterministic behavior of channel use in order to predict future PU occupation. The authors reduce the complexity of the analysis and the amount of memory storage needed by assuming a sequence of binary states, thus also simplifying spectrum occupation ("1" for empty, "0" for in use). In the tests that were conducted, the short range prediction factor is highly satisfactory for the first two tests; however, in the third sample, the success of the prediction is strongly degraded because the model does not update itself and, in addition, the behavior of the data is not deterministic. In theory, this problem could be resolved by increasing the order at the expense of an exponential increase of the parameters for generating the prediction. From a deterministic perspective, the estimation is quite solid for the first four time slots according to tests conducted for three different bands in a GSM network during a capture with duration of 17 ms.

The article in Uyanik et al. (2012) proposes three prediction mechanisms based on correlation, linear correlation and regression, and self -correlation, based on previous decisions, to predict future spectrum status as well as decision making regarding PU occupation. The prediction-based correlation scheme uses the Pearson Correlation Coefficient which is measured from historical samples of windows; if the coefficient is above a certain threshold, the prediction window is filled with the latest sample. Linear regression prediction based on the Pearson Coefficient establishes the correlation between the spectrum detection status and the index vector. This coefficient is determined by a threshold value similar to the previous approximation, and there will be a correlation and regression if a linear relationship exists. If a relationship exists, it's used for spectrum prediction. Simulations show the proposed prediction scheme exhibits better results in diverse simulation settings. Furthermore, in order to obtain a more realistic evaluation, it's necessary to take into account the values of the utility system along with the PU's disturbance relationship values.

2.4 Characterization algorithms based on other methodologies

In Ghosh et al. (2010), a spectrum occupation model is proposed. It was designed to accurately generate both the time-based and the frequency-based behavior of several wireless

transmissions. Using the statistical characteristics of real radio frequency measurements, first- and second-order parameters are obtained. They were used in a statistical spectrum occupation model based on a combination of several different probability density functions. The characteristics of the outputs of the proposed spectrum occupation model were compared with spectrum measurements obtained from real-time frequency measurements in the 928–948 MHz band. The article's main contribution is the validation of the spectrum occupation model developed to predict the arrival of PUs in the operating spectrum.

In Masonta (2013), it is concluded that there is a significant increase in the proposals for modeling PU activity as Poisson processes with exponentially distributed inter-arrivals. This is the case in Lee and Akyildiz (2011), where an adaptative decision framework is proposed for finding spectrum band groups that meet secondary user requirements by characterizing the wireless spectrum as a function of PU activity and the sensing stage. However, based on a study conducted in Canberk et al. (2011), it is concluded that the duration of primary node activity does not exhibit an exponential behavior, and the high fluctuations of PUs could possibly not be detected, in breach of the Poisson assumption, thus reducing the validity to constant flows without peaks.

A solution to the issue with Poisson processes (posed previously) in the characterization of PUs is discussed in Canberk et al. (2011). An algorithm based on first-difference filter clustering and correlation was implemented to represent primary user activity. The essential contribution is that characterizing variables captured in the environment through filtering results in a more accurate and appropriate use of the spectrum that improves CRN performance, in contrast with the approximation obtained through Poisson.

The state of the art described in Masonta (2013) can be synthesized by saying that since there is no guarantee that a band is available during the period required by a SU for transmission, it's important to take into account how frequently PUs appear. Using CR's learning ability, a PU's history of spectrum use activity can be used to predict the spectrum's future profile, a process that is achieved through characterization. SUs can decide on the best spectrum bands available for transporting their data considering the future behavior of PUs. The above statement reflects this article's intention, which is to characterize PUs using a LSTM neural network methodology that includes the Deep Learning concept Graves et al. (2013).

3 Development of the proposal

Making highly accurate predictions is quite beneficial to planning and control in many fields of research and development. However, an elevated accuracy level entails a high level of difficulty Salgado (2014); however, there are promising prediction techniques applicable to CR and based on AI that have the ability to provide consciousness, reasoning and learning He et al. (2010); such elements are capable of interacting, thus benefiting the capability for autonomy of cognitive radio networks and therefore raising performance with a low level of difficulty that is appropriate to CR's autonomous learning needs, as is the case for LSTM.

Future channel state estimation in the GSM and WiFi bands (from the PU perspective) was addressed as a binary series prediction problem, based on the conversion of power levels (dBm) captured and delivered by the spectrum analyzer to discrete values (Sect. 3.4), and using a Deep Learning based recurrent neural network. To that end, the theoretical concept of LSTM is defined in this section; also defined is the way in which the PU input signal for the system is modeled, and the LSTM network's layered structure is described; after that, the mathematical model, explaining the LSTM system proposed for the characterization and

implemented in C#, is built, describing the existing interaction among input neurons, memory cells (hidden layers), and output neurons during the training or learning process.

3.1 Long short-term memory

Traditional artificial neural networks are not capable of storing information. In order to do so, it's necessary to modify the topology by creating recurrent structures that retro feed the neurons and allow information storage. Such structures are known as recurrent neurons. A set of such neurons is called a recurrent neural network (RNN). An RNN allows storage of subsequent states in different time intervals where the parameters are shared among the different parts of the model, which allows for better generalization Veeriah et al. (2015). One of the problems of an RNN is long-term dependency, which suggests the need to not always study historical data to perform a current task. This implies an RNN stores only information learned in the past, and is not capable of storing new information in the short term. LSTM can be expressly designed to avoid the long-term dependency problem by remembering information during long periods of time and learning new information in the present. LSTM blocks contain memory cells which allow a value to be remembered for an arbitrary time period and used when necessary. There is also a forget layer which can erase memory content that is not useful. All the components are built for differentiable functions and are trained during the backpropagation process Artiemjew and Jiao (2015). The structure of an LSTM can be represented as shown in Fig. 3, where the memory cell is identified with a letter 'C', the forget layer, with an 'O', the input layer, with an 'E', and the output layer, with an 'S'.

3.2 Modeling of the input signal and LSTM model layers

A discrete input signal indicates the presence (1) or absence (0) of a PU within a spectrum band for a time period T, according to Eq. (1), in which, based on the binary sequence, the predictor is trained to forecast channel status not only in the next time slot, but also in subsequent points in time based on the historical data of a PU's behavior in the channel.

$$X_0^T = \left[x_0, x_1, x_2, x_{3,\ldots,}x_T\right] \tag{1}$$

Determining the exact number of neurons for resolving the characterization problem is especially difficult. A very small neural network cannot learn how to correctly solve the problem, but a very large network will generate an over adjustment (i.e. the problem is singled out, not generalized) Kwok and Yeung (1997). In addition, it must be taken into account that the more layers and neurons there are, the training time is greater, and more resources are used. In this article, the authors used a numerical optimization technique based on the geometric pyramid rule, which is especially useful when the number of input layer neurons is greater than the number of output layer neurons Masters (1993), as is the case with this problem. Since it's necessary to divide the number of input layer neurons n times a power of 2 until one is obtained, Eq. 2 is arrived at, where Co corresponds to the number of input layer neurons, and n to the number of existing layers.

$$1 = \frac{Co}{2^n} \Rightarrow n = \lceil log_2\,(Co)\rceil \tag{2}$$

It's possible to discern from the preceding equation that the number of layers grows in a controlled fashion according to the increase in the number of input neurons. Due to the fact that a design decision was made to develop a dynamic software application where the creation of the LSTM neural network varies and depends on the input sequence, the total number of

Table 1 Notation for the development of the mathematical model

	Memory block	Input gate	Forget gate	Output gate	Memory cell
Subindex	i	l	$\emptyset$	w	c
Input	x_i	a_l^t	$a_\emptyset^t$	a_w^t	$a_c^t, \quad s_c^t$
Output	y_i	b_l^t	$b_\emptyset^t$	b_w^t	$b_c^t = b_w^t\left(s_c^t\right)$
Number of units	I	N/A	N/A	N/A	C
Activation function	N/A	f Sigmoid function	f Sigmoid function	f Sigmoid function	f (in-cell) h (out-cell)

neurons that comprise a network topology is obtained from Eq. 3.

$$N = \sum_{i=0}^{Co} \lceil \frac{Co}{2^i} \rceil \tag{3}$$

Approximating the preceding series Eqs. 3, 4 is obtained.

$$N \approx Co\left(2 - 2^{-Co}\right) \tag{4}$$

Taking Co (from Eq. 4) as a very large number, it can be assumed that the total number of neurons tends to:

$$\lim_{Co \to \infty} Co\left(2 - 2^{-Co}\right) = 2Co = \infty \tag{5}$$

Equation 5 indicates that as the number of input layer neurons increases, the total number of neurons is approximately twice the number of input layer neurons.

3.3 LSTM system operating model

LSTM can be considered a differentiable function approximator that is usually trained with a descending gradient Graves (2012). Although a truncated form of BPTT (Backpropagation Through Time) was initially employed to approximate the error gradient Hochreiter and Schmidhuber (1997), a BPTT calculation without truncation, based on the discussion by Graves in Graves and Schmidhuber (2005), was used in this research. The operation of the LSTM neural network described in Sects. 3.3.1 and 3.3.2 uses the notation shown in Table 1 (which is in accordance with the description in Graves 2012).

3.3.1 Forward pass equations

For the three cell gates (input, output and forget) Graves (2012), the propagation functions a_l^t, $a_\emptyset^t$ and a_w^t take into account not only the weighted sum of the current inputs, but also the outputs for the immediately preceding time for the blocks in the hidden layer and the status of other cells in the same block (except in the output gate where the current status of the cells is required). In this regard, Eq. 6 through 11 Graves (2012) result from the analysis of the LSTM block (Fig. 4 modified from Palangi et al. 2015) for each gate and memory cell comprising the model.

For the input gate:

$$a_l^t = \sum_{i=1}^{L} w_{il}^t x_i^t + \sum_{h=1}^{H} w_{hl}^t b_h^{t-1} + \sum_{c=1}^{C} w_{cl} s_c^{t-1} + \theta_l \tag{6}$$

$$b_l^t = f(a_l^t) \tag{7}$$

For the forget gate:

$$a_\emptyset^t = \sum_{i=1}^{L} w_{i\emptyset}^t x_i^t + \sum_{h=1}^{H} w_{h\emptyset}^t b_h^{t-1} + \sum_{c=1}^{C} w_{c\emptyset} s_c^{t-1} + \theta_\emptyset \tag{8}$$

$$b_\emptyset^t = f(a_\emptyset^t) \tag{9}$$

For the output gate:

$$a_w^t = \sum_{i=1}^{L} w_{iw}^t x_i^t + \sum_{h=1}^{H} w_{hw}^t b_h^{t-1} + \sum_{c=1}^{C} w_{cw} s_c^{t-1} + \theta_w \tag{10}$$

$$b_w^t = f(a_w^t) \tag{11}$$

In order to describe a cell's behavior, two elements must be taken into account. The first one is the a_c^t propagation function, which depends not only on current inputs, but also on outputs for the immediately preceding time from the other blocks in the hidden layer. The second one is the s_c^tneuron status, which indicates if the neuron is keeping the information or will forget it, and depends on the outputs from the forget gate and the input gate.

The neuron output b_c^t will indicate if new learning was generated or the stored information is kept. Now that the above is clear and based on Fig. 4, it is concluded that cell status and output are given by Eqs. 12–14 Graves (2012).

Neuron status:

$$a_c^t = \sum_{i=1}^{L} w_{ic}^t x_i^t + \sum_{h=1}^{H} w_{hc}^t b_h^{t-1} \tag{12}$$

$$s_c^t = b_\emptyset^t s_c^{t-1} + b_l^t g\left(a_c^t\right) \tag{13}$$

Neuron output:

$$b_c^t = b_w^t h\left(s_c^t\right) \tag{14}$$

3.3.2 Backward Pass equations

In order to obtain the Backward Pass equations, the BPTT method is used (as previously mentioned) Graves (2012), which implies using the chain rule to calculate the derivatives of errors at the exit of the components of an LSTM block.

When defining the outputs for input gate, output gate and forget gate as a_j^t, they can be represented as described in Eq. 15.

$$\delta_j^t = \frac{\partial E}{\partial a_j^t}, \text{ where } j \in \{l, \emptyset, w\}. \tag{15}$$

In addition, in defining cell output (ϵ_c^t) and cell status (s_c^t):

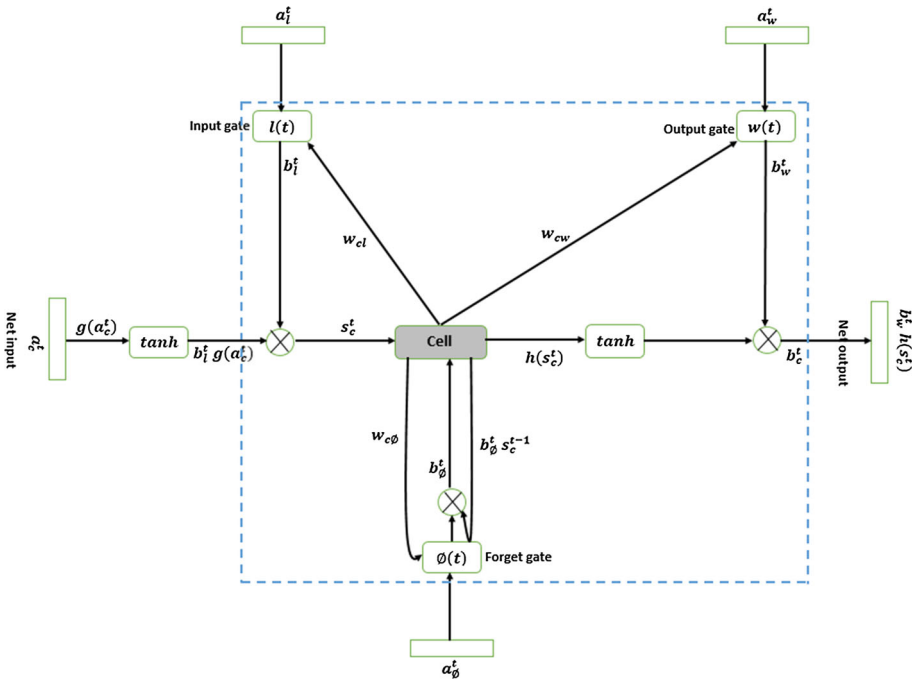

Fig. 4 LSTM architecture used for the characterization of PUs [modified of Palangi et al. 2015]

$$\epsilon_c^t = \frac{\partial E}{\partial b_c^t} \tag{16}$$

$$\epsilon_s^t = \frac{\partial E}{\partial s_c^t} \tag{17}$$

Defining E (in Eqs. 16, 17) as the loss function (error), and based on the fact that the purpose is to establish how the error varies when the weights are modified, the following is obtained based on the chain rule:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial w_{ij}} = b_i \frac{\partial E}{\partial a_j} \tag{18}$$

From Eq. 18, it's clear the goal is to calculate $\frac{\partial E}{\partial a_j}$, but taking into account that in the case of LSTM, 4 types of a exist, namely ($\frac{\partial E}{\partial a_w^t}$) output gate, ($\frac{\partial E}{\partial a_c^t}$) cells, ($\frac{\partial E}{\partial a_\emptyset^t}$) forget gate, and ($\frac{\partial E}{\partial a_l^t}$) input gate, all of which can be defined as shown in Eqs. 19–22.

$$\frac{\partial E}{\partial a_w^t} = \sum_{c=1}^{C} \frac{\partial E}{\partial b_c^t}\frac{\partial b_c^t}{\partial b_w^t}\frac{\partial b_w^t}{\partial a_w^t} = \frac{\partial b_w^t}{\partial a_w^t}\sum_{c=1}^{C} \frac{\partial E}{\partial b_c^t}\frac{\partial b_c^t}{\partial b_w^t} \tag{19}$$

$$\frac{\partial E}{\partial a_c^t} = \frac{\partial E}{\partial s_c^t}\frac{\partial s_c^t}{\partial a_c^t} \tag{20}$$

$$\frac{\partial E}{\partial a_\emptyset^t} = \sum_{c=1}^{C} \frac{\partial E}{\partial s_c^t}\frac{\partial s_c^t}{\partial b_\emptyset^t}\frac{\partial b_\emptyset^t}{\partial a_\emptyset^t} = \frac{\partial b_\emptyset^t}{\partial a_\emptyset^t}\sum_{c=1}^{C} \frac{\partial E}{\partial s_c^t}\frac{\partial s_c^t}{\partial b_\emptyset^t} \tag{21}$$

$$\frac{\partial E}{\partial a_l^t} = \sum_{c=1}^{C} \frac{\partial E}{\partial s_c^t} \frac{\partial s_c^t}{\partial b_l^t} \frac{\partial b_l^t}{\partial a_l^t} = \frac{\partial b_l^t}{\partial a_l^t} \sum_{c=1}^{C} \frac{\partial E}{\partial s_c^t} \frac{\partial s_c^t}{\partial b_l^t} \tag{22}$$

Taking into account that the summation is done over c because the model is developed in a single block (with C cells inside), the mathematical descriptions shown in Eq. 23 are found when calculating the respective derivatives Graves (2012).

$$\begin{aligned}
&\frac{\partial s_c^t}{\partial b_l^t} = g\left(a_c^t\right) \quad \frac{\partial s_c^t}{\partial b_\emptyset^t} = s_c^{t-1} \quad \frac{\partial b_c^t}{\partial b_w^t} = h\left(s_c^t\right) \\
&\frac{\partial b_\emptyset^t}{\partial a_\emptyset^t} = f'\left(a_\emptyset^t\right) \quad \frac{\partial s_c^t}{\partial a_c^t} = \quad b_l^t g'\left(a_c^t\right) \quad \frac{\partial b_l^t}{\partial a_l^t} = f'\left(a_l^t\right) \\
&\frac{\partial b_w^t}{\partial a_w^t} = f'\left(a_w^t\right)
\end{aligned} \tag{23}$$

Based on the mathematical analysis applied above, the following Backward Pass equations are obtained Graves (2012):

Output gate:

$$\delta_w^t = \frac{\partial E}{\partial a_w^t} = f'\left(a_w^t\right) \sum_{c=1}^{C} \epsilon_c^t h\left(s_c^t\right) \tag{24}$$

Cell:

$$\delta_c^t = \frac{\partial E}{\partial a_c^t} = \epsilon_s^t b_l^t g'\left(a_c^t\right) \tag{25}$$

Forget gate:

$$\delta_\emptyset^t = \frac{\partial E}{\partial a_\emptyset^t} = f'\left(a_\emptyset^t\right) \sum_{c=1}^{C} \epsilon_s^t s_c^{t-1} \tag{26}$$

Input gate:

$$\delta_l^t = \frac{\partial E}{\partial a_l^t} = f'\left(a_l^t\right) \sum_{c=1}^{C} \epsilon_s^t g\left(a_c^t\right) \tag{27}$$

Note that Eqs. 24–27 depend on the ϵ_s^t and ϵ_c^t terms; therefore it's necessary to determine how the error is affected when making changes both to cell outputs and cell status.

In this case, keep in mind that the error is a function with variables that are the K outputs generated by the H blocks of the hidden layer; in fact, for a given block, the resulting output in a time t, will affect the K units of the output layer (at a time t) and at the next input to each one of the H blocks in the hidden layer. Therefore, ϵ_c^t can be defined as:

$$\epsilon_c^t = \frac{\partial E}{\partial b_c^t} = \sum_{k=1}^{K} \frac{\partial E}{\partial a_k^t} \frac{\partial a_k^t}{\partial b_c^t} + \sum_{h=1}^{H} \frac{\partial E}{\partial a_h^{t+1}} \frac{\partial a_h^{t+1}}{\partial b_c^t} \tag{28}$$

The cell output is described as follows by Eq. 29:

$$\epsilon_c^t = \sum_{k=1}^{K} \frac{\partial E}{\partial a_k^t} w_{ck} + \sum_{h=1}^{H} \frac{\partial E}{\partial a_h^{t+1}} w_{ch} \tag{29}$$

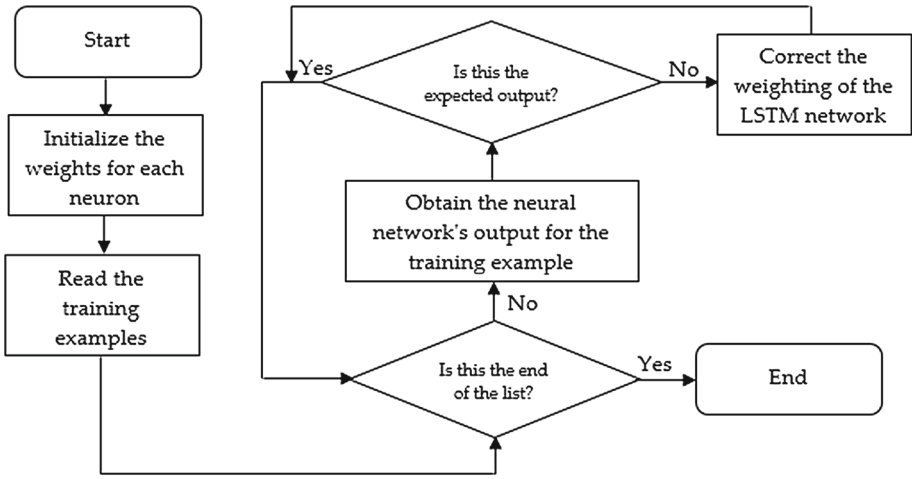

Fig. 5 Flow chart for LSTM training

Finally, it's necessary to analyze what happens with the error if changes to cell status are made. The status of the cell c in time s_c^t, indicates whether or not the information stored at the time was modified; therefore, s_c^t is a value that affects the inputs of all gates, the next status of the cell and, obviously, the output of the cell. This is mathematically expressed in Eq. 30.

$$\epsilon_s^t = \frac{\partial E}{\partial s_c^t} = \frac{\partial E}{\partial b_c^t}\frac{\partial b_c^t}{\partial s_c^t} + \frac{\partial E}{\partial s_c^{t+1}}\frac{\partial s_c^{t+1}}{\partial s_c^t} + \frac{\partial E}{\partial a_l^{t+1}}\frac{\partial a_l^{t+1}}{\partial s_c^t} + \frac{\partial E}{\partial a_\emptyset^{t+1}}\frac{\partial a_\emptyset^{t+1}}{\partial s_c^t} + \frac{\partial E}{\partial a_w^t}\frac{\partial a_w^t}{\partial s_c^t} \quad (30)$$

The status of the cell is Graves (2012): (Eq. 31)

$$\epsilon_s^t = \epsilon_c^t\frac{\partial b_c^t}{\partial s_c^t} + \epsilon_s^{t+1}\frac{\partial s_c^{t+1}}{\partial s_c^t} + \delta_l^{t+1}\frac{\partial a_c^{t+1}}{\partial s_c^t} + \delta_\emptyset^{t+1}\frac{\partial a_\emptyset^{t+1}}{\partial s_c^t} + \delta_w^{t+1}\frac{\partial a_w^{t+1}}{\partial s_c^t} \quad (31)$$

3.4 Flowchart and pseudo-code for the LSTM system

The training flowchart (Fig. 5) for the process begins by randomly initializing each neuron with values between -1 and 1; after that, each training example is read and the output is compared with the expected output. If the response obtained does not match what was expected, the algorithm calculates the error between the system output and the expected output, correcting each weight of the gates (input, output, forget) and the cell by applying weighting and making use of hyperbolic tangent and sigmoid functions until completing all training examples, thus bringing the model's output close to the expected output (by error reduction, as shown in Sect. 3.3).

Part of the pseudo-code for the algorithm that was implemented is shown below.

LSTM algorithm

```
Data: The existence of a Wo, Wf, Wi and Wc array that represents the neural network
Result: A neural network trained with data from the training examples
forgetLayer = Wf.size(); //The size of the array representing the neural network is obtained.
for  i = 0; i < neurons; i++ do
      Wf[i] = random(-1,1); //Each layer of the neural network is initialized.
      Wi[i] = random(-1,1);
      Wc[i] = random(-1,1);
      Wo[i] = random(-1,1);
end
bf = 0.5; //Approximation of the output obtained for each layer.
bc = 0.5;
bi = 0.5;
bo = 0.5;
inputs = readInputs(); //Input examples are read.
outputs = readOutputs(); //Output examples are read.
size = inputs.size(); //The size of the examples is obtained.
for       i = 0; i < size; i++ do
      sumf = 0;
      sumi = 0;
      sumc = 0;
      sumo = 0;
      for j = 0; j < neurons; j++ do
            sumf = sumf + Wf[j]*inputs[i][j]; //The output for each example in each layer is calculated.
            sumi = sumi + Wi[j]*inputs[i][j];
            sumc = sumc + Wc[j]*inputs[i][j];
            sumo = sumo + Wo[j]*inputs[i][j];
      end
      ft = sigmoid(sumf + bf); //Approximations are made for each network output.
      it = sigmoid(sumi + bi);
      dct = tanh(sumc + bc);
      ct = ft + it*dct;
      ot = sigmoid(sumo + bo);
      output = ot*tanh(ct); //The neural network output is calculated.
      if  output != outputs[i] then
        error = outputs[i] - output;
     for  j = 0; j <neurons; j++ do
        Wo[j] = Wo[j] + inputs[i][j]*e; //Each neuron is traversed and the weighting is corrected
         with respect to the calculated error.
     end
     bo = 0.5 + error; //The shift is corrected.
end
```

4 Results analysis and evaluation

For the evaluation of the performance of the PU characterization algorithm (consisting of the white spaces in which licensed users will not use the available spectrum so that the white spaces can be used in an opportunistic fashion by cognitive nodes or SUs), the LSTM net-

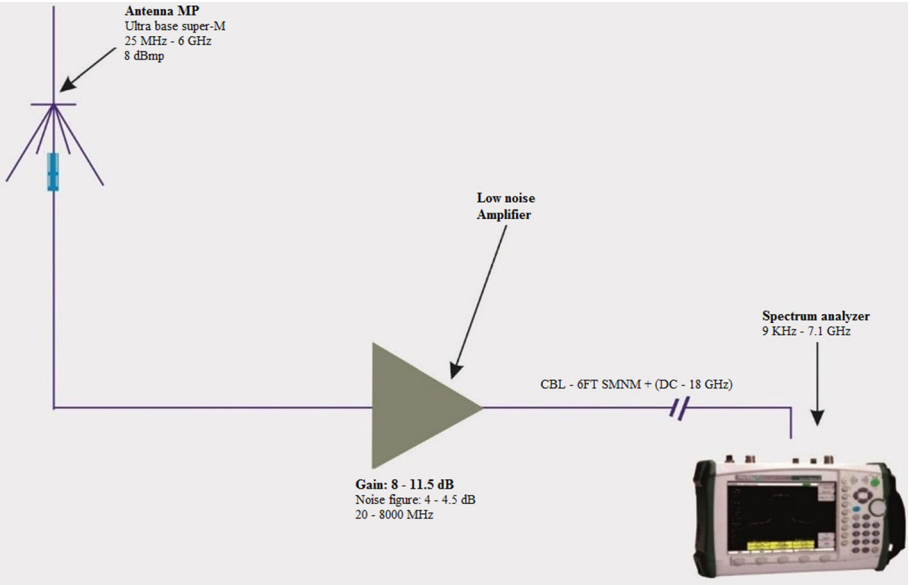

Fig. 6 Interconnection of equipment for capturing spectrum occupation data Hernández et al. (2015)

work was subjected to two test cases; in the first case, the performance of the modeling and prediction system was analyzed when the input signal for the LSTM network was computationally generated for different behavior patterns (see Sect. 4.2), which are hard to find in conventional wireless networks; in the second case, the forecast produced by the algorithm for real data sequences captured in the GSM-850 and WiFi bands (see Sect. 4.3) was studied based on the premise that, out of 100% of the data used, 70% is used in the training stage of the LSTM network, and the remaining 30%, for validation (prediction estimation).

4.1 Capture and processing of spectrum information

For data capture, the first step was to determine what wireless network application would be used to evaluate the Deep Learning based technique Hochreiter and Schmidhuber (1997). Cellular (GSM) and Internet access (WiFi) communications were chosen as the main objective. The second step was to select the spectrum detection technique to be used. Energy detection was selected because it's easily implemented and has low requirements Hernández et al. (2015). The manner in which data capture was performed is shown in Fig. 6; Table 2 shows the spectrum measurement technical specifications. Table 3 shows the characteristics of the cluster used as a computing resource for the development of the algorithm and the execution of the training and prediction tests.

For the processing of spectrum information, measurements were made every 290 milliseconds in the WiFi band (2.4 GHz), and GSM (824.8 MHz uplink, 869.8 MHz downlink) in terms of transmission power; in addition, in order to facilitate pattern recognition, power levels were presented in binary form based on the definition established in Eq. 32,

$$f(x) = \begin{cases} 0, & if\ x \leq a \\ 1, & if\ x > a \end{cases} \tag{32}$$

where the values of a are -89 dBm for GSM, and -88 dBm for WiFi.

Table 2 Specifications of equipment for spectrum measurement and capture

Equipment	Specifications	
	Frequency range	Model reference
Spectrum analyzer	9 KHz–7.1 GHz	MS2721B Anritsu
Discone antenna	25 MHz–6 GHz	Super-M ultra base
Low noise amplifier	20 MHz–8 GHz	ZX60 - 8008 -S+
Broand cable	DC–18 GHz	CBL-6FT SMNM+

Table 3 Cluster specifications

Feature	Description
Equipment and brand	KVM Virtual Machine - BIOS Openstack Foundation 2015.1
Brand	DELL R900 server
Number of processors	Intel(R) Xeon(R) CPU E7450 @ 2.40 GHz, 24 Cores
RAM	64 GB DDR2
Storage system	1000 GB ext4
Operating system	Ubuntu server 14.04.04 with an XFCE4 desktop environment

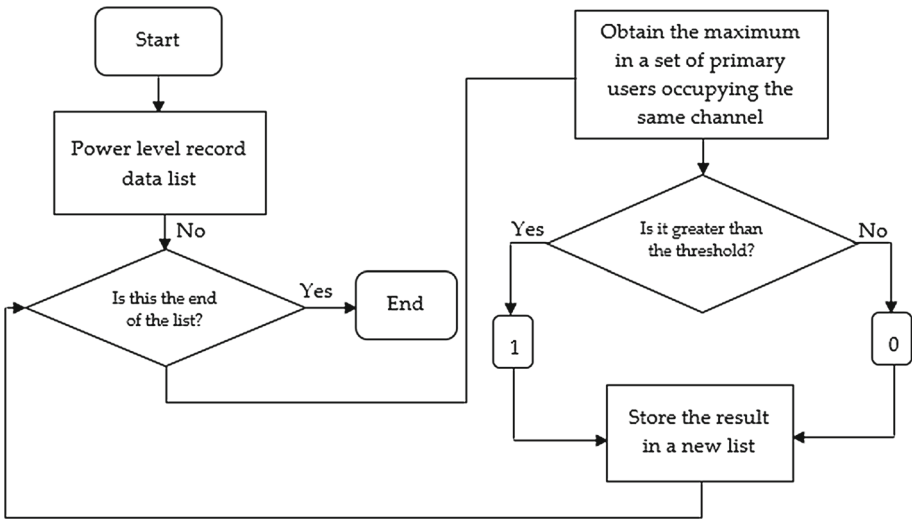

Fig. 7 Flowchart for the discretization of spectrum data

Figure 7 shows the procedure for converting the spectrum data traces into discrete signals. It's worth noting that when performing the tests, a 6.79 GB database was available with information on GSM traffic traces, and a 9.63 GB database was available for WiFi traces, with more than 10,000 data records per files obtained from Pedraza et al. (2016).

4.2 First group of test cases

Behavior patterns (of multiple sizes) were created through simulation based on what's suggested in Saleem and Rehmani (2014) and in accordance with Table 4

Table 4 Test cases for PU traffic traces generated through simulation

Identifier	Test case	Description
TC1	i % 2 === 0	Refers to historical data where all even-numbered time units show channel occupation
TC2	i % 5! === 0	Refers to historical data where all time units that are not a multiple of 5 show channel occupation
TC3	i % 3 === 0	Refers to historical data where all time units that are a multiple of 3 show channel occupation
TC4	i % 3 === 0 and i % 2 === 0	Refers to historical data where all time units that are a multiple of 3 and 2 show channel occupation
TC5	Random	Refers to historical data where channel occupation is randomly generated

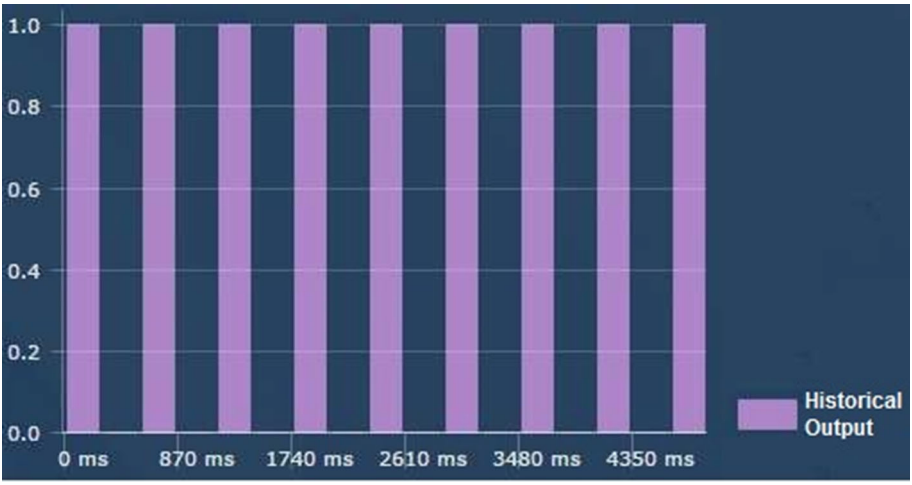

Fig. 8 Behavior of historical data for 77 samples

For qualitative purposes, results are presented for the LSTM algorithm when modeling and estimating the future behavior of a licensed user (for TC1), with a high fluctuation between presence and absence in the licensed channel Saleem and Rehmani (2014). The binary sequence that simulates channel use is made up of 77 digits. Figure 8 shows the sequence for the first 17 digits as 10101010101010101, where PU presence is represented by a 1, and PU absence, by a 0.

The application that was developed generates adaptively (Fig. 9) the LSTM neural network structure that is most appropriate for the input sequence according to what was set forth in Sect. 3.2.

The learning stage (training modeling) is shown in Fig. 10, where it is concluded that the LSTM network was 100% capable of determining the pattern of channel use.

From the future estimation (prediction projection) provided by the neural network (Fig. 11), it's worth highlighting that the accuracy level between the original signal (purple-colored sequence) and the system-projected signal (blue lines) is 100%, which suggests the prediction error is 0%, thus indicating that the neural system is very efficient for the case

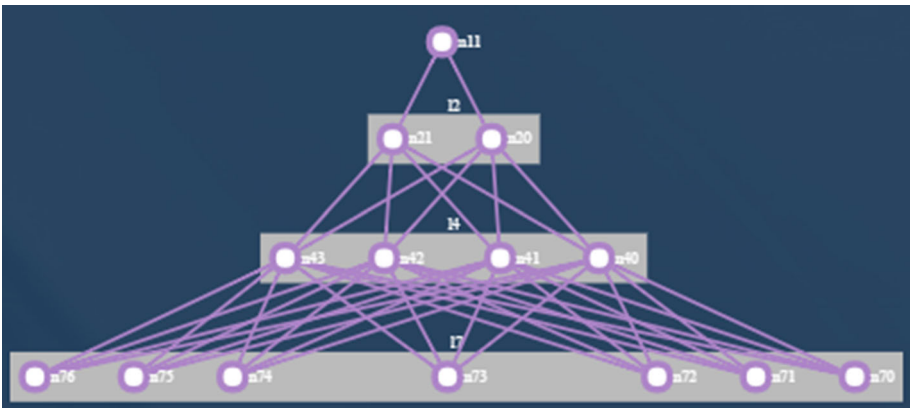

Fig. 9 Neural network topology

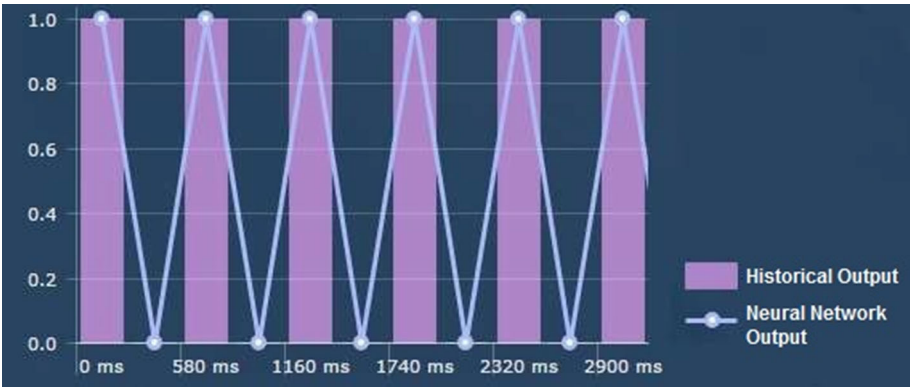

Fig. 10 Results of the training stage (network learning or modeling stage)

under evaluation. Quantitative results for the different cases listed in Table 4 are shown in Table 5.

The performance evaluation metrics (Table 5) refer to average values, because historical data of various sizes was created (17, 35, 77, 157 and 200 binary digits), applying 10 tests for each case because different solutions could be obtained each time the algorithm is executed. The LSTM algorithm was validated with the same metrics and the same considerations, but using a pyramid multilayer perceptron neural network (MLPNN) (See Table 6).

Analysis of Tables 5 and 6 reveals that the average prediction error for LSTM ranges from 0 to 37.48%, placing the forecast level above 62.50% in the worst case (TC5), a percentage that is higher than what was found with MLPNN (47.33%). This indicates LSTM was able to generalize the behavior for the various cases that were submitted and was able to adequately predict PU behavior at any instant in time t as long at the PU continues to have the same behavior. Another important feature is that although LSTM has more neurons in its structure than MLPNN, it required less iterations in TC1 through TC4, which proves that the complexity of the LSTM structure allows abstracting the PU signal behavior pattern at a lower computing cost when the size of the matrix used for historical data has a short length. Finally, the average

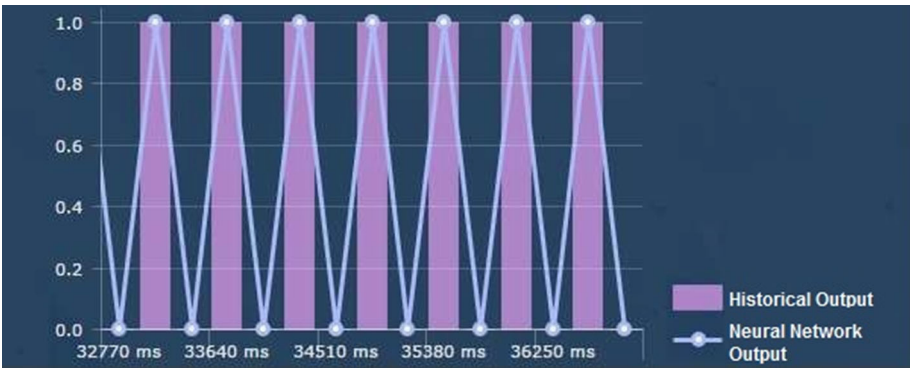

Fig. 11 Prediction results

Table 5 Performance of LSTM in the characterization of PUs

Test case	Average validation error (%)	Average prediction error (%)	Number of iterations	Processing time
LSTM				
TC1	0.087583	0	1352	54.8
TC2	0.0936039	20.046595	1422	75.6
TC3	0.8194522	28.9938556	5030	549.1
TC4	0.7566375	15.8806964	3757	820.4
TC5	0.798135	37.4858167	17,402	6758.6

Table 6 Performance of MLPNN in the characterization of PUs

Test case	Average validation error (%)	Average prediction error (%)	Number of iterations	Processing time
MLPNN				
TC1	0.04994593	0	1594	78.9
TC2	0.06868485	24.9820789	4314	244.1
TC3	0.105482774	37.6014337	5491	517.2
TC4	0.8826215	20.8151562	4702	943.1
TC5	0.5043697	52.6732207	5139	2263.5

validation error is very small for both types of neural networks, a condition that guarantees the network can be optimally modeled.

4.3 Second group of test cases

To demonstrate the viability of the proposed algorithm with real GSM and WiFi traffic traces (according to the characteristics laid out in Sect. 4.1), a metric called Index of Occupation (*Io)* was defined (Eq. 33) in order to divide spectrum band use levels into high, medium and low; this allows a more objective and detailed assessment.

$$Io = \frac{\sum_{x=0}^{n} t\,(x)}{n} 100\% \tag{33}$$

where $t\,(x)$ corresponds to discretized data flows, and n is the number of elements at $t\,(x)$.

This case study also includes the results of modeling and predicting the chaotic behavior of PUs in the bands by applying the ANFIS algorithm. The resulting outputs are summarized in Tables 7 and 8. For reference, a trace with 20,000 data was used to feed the system for each one of the three frequency bands selected (according to their occupation index), and 10 tests were executed in each case.

Evaluation of the metrics for each case suggests, first, that processing time is higher with LSTM due to the larger size of the traces and also because this type of recurrent network uses memory cells to store information about patterns that were found and could be reused in the future. The capability to store and forget patterns directly affects the Validation Error variable, which is much better in LSTM than MLPNN.

It was also observed that the Training Error is lower in LSTM because of its greater ability for pattern recognition thanks to the use of forget, input and output gates, and memory cells.

Regarding LSTM's accuracy percentage, values range between 97.09% (for a low) and 77.14% (for a high Io) in GSM systems, and between 87.25% (for a low Io) and 63.82% (for a high Io) in WiFi systems, which validates that LSTM is more efficient than MLPNN and ANFIS; however, it's important to note that greater efficiency implies greater hardware requirements, a factor that isn't relevant if the prediction system is implemented in cognitive radio networks with a centralized topology.

When examining the success percentages of PU arrival/non-arrival in the spectrum bands (Fig. 12), it can be concluded that, for the occupation indices defined in Tables 7 and 8, and using as reference the algorithms that exhibited the best performance, the prediction success average with LSTM is 87.34% for GSM and 76.30% for WiFi, while the efficiency achieved with ANFIS is 86.68% for GSM and 72.62% for WiFi.

Also observed is a decreasing linear trend with a greater slope in forecasts as spectrum occupation increases, due to greater intermittence in spectrum use by licensed users.

Lastly, better performance in PU characterization (for the three methodologies analyzed - LSTM, MLPNN and ANFIS) was observed in the GSM spectrum band, probably due to the more chaotic nature of WiFi traffic flows.

5 Discussion

In the AI field, neural networks have been extensively applied to time series given the prediction capabilities for unknown time units, due to the ability to be trained by means of examples in order to abstract a behavior. This contrasts with other AI techniques, which obtain the knowledge from an expert through the representation of variables relevant to the solution of the problem. One of the most widely used supervised learning methodologies for the characterization of PUs is multilayer neural networks (MLPNN), which can achieve an efficiency improvement of up to 60% in prediction, as concluded in Adeel et al. (2014) (although higher percentages were achieved in tests); however, there have been recent proposals for the use of techniques based on Deep Learning, due to its high abstraction level Kalkan (2015) for the solution of multiple problems Sun et al. (2016), Palangi et al. (2016), Sundermeyer et al. (2016), Gers and Schmidhuber (2001), a significant reason for suggesting its use in CR.

Table 7 Algorithm performance for GSM flows

Metric	LSTM			MLPNN			ANFIS		
	High index	Medium index	Low index	High index	Medium index	Low index	High index	Medium index	Low index
Number of iterations	5000	5000	5000	5000	5000	5000	5000	5000	5000
Training error	0.4969	0.3711	0.1091	0.5163	0.3866	0.1485	0.507	0.3771	0.1295
Processing time (ms)	7,527,099	7,395,395	7,472,671	1,735,304	1,665,314	1,630,825	4,279,211	4,011,087	3,997,590
Correlation coefficient	0.77	0.85	0.94	0.62	0.70	0.78	0.65	0.74	0.82
Validation error (%)	20.21	12.52	3.35	21.71	14.02	4.85	20.68	13.54	3.99
Prediction error (%)	22.86	12.20	2.91	24.36	13.70	4.41	23.44	12.82	3.70

Table 8 Algorithm execution results for WiFi flows

Metric	LSTM			MLPNN			ANFIS		
	High index	Medium index	Low index	High index	Medium index	Low index	High index	Medium index	Low index
Number of iterations	5000	5000	5000	5000	5000	5000	5000	5000	5000
Training error	0.7017	0.5520	0.3091	0.7970	0.5602	0.3588	0.7411	0.5587	0.3274
Processing time (ms)	6,777,423	6,830,459	6,700,412	1,454,489	1,493,626	1,400,650	4,092,147	3,896,192	3,701,767
Correlation coefficient	0.62	0.75	0.85	0.43	0.65	0.73	0.51	0.67	0.78
Validation error (%)	38.35	21.49	10.23	41.51	23.02	11.28	39.27	22.06	10.74
Prediction error (%)	36.18	22.16	12.75	48.14	24.59	16.88	44.14	23.12	14.88

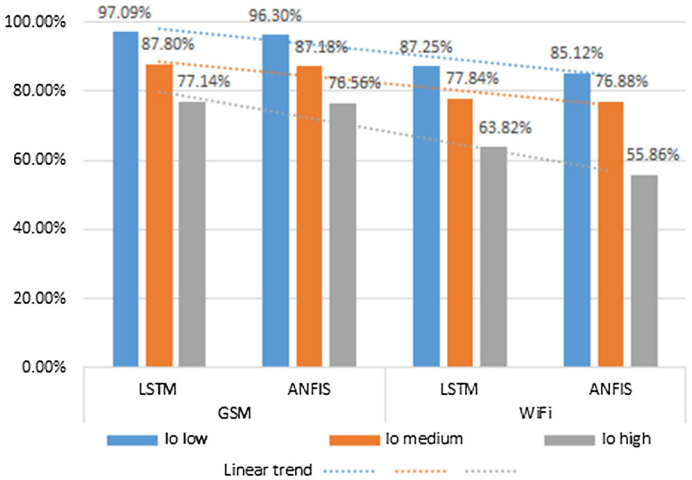

Fig. 12 Prediction accuracy percentage for LSTM and ANFIS

From the analysis in the previous section, it's evident that although LSTM exhibits a greater prediction capability, it still has a significant estimation error in cases in which PU behavior is chaotic or random; however, obtaining an error close to zero is difficult due to signal nature, a condition that can be supported from the perspective of entropy. From Eq. 34, when entropy is 1, this indicates there is a 50% probability the spectrum band is occupied at any point in time, which generates a high level of uncertainty when making channel occupation estimations. The opposite occurs when the value tends to zero (a more favorable condition).

$$E_s = \sum_{i=0}^{n} p\left(x_i\right) * log_2\left(\frac{1}{p_{x_i}}\right) \tag{34}$$

where $p\left(x_i\right)$ refers to the probability of appearance of the character x_i and n to the number of characters.

Based on the above consideration, when calculating (for example) values for GSM-LSTM with high, medium and low occupation indices, values of 0.7317737, 0.5529701 and 0.1979427 respectively are obtained, which is coherent with the prediction errors in Table 7; on the other hand, historical data generate indications of PU behavior, but no guarantee that it will actually occur again. However, having an indication of possible behavior allows a cognitive network central station to be prepared to take actions on the possible assignment of a frequency band to a SU.

An additional indicator for verifying which is the best model among the LSTM, MLPNN and ANFIS algorithms is the correlation coefficient (obtained by averaging the values of the variable Io) in Tables 7 and 8, and shown in Fig. 13; a value close to 1.0 in the case of LSTM makes it possible to conclude LSTM is the best of the three models that were evaluated.

A final contribution of the application that was developed (for the LSTM algorithm) is the ability to automatically create a neural structure according to the size of the trace to be characterized; this is positive because no additional efforts are required to build the topology when modifying the behavior of input data. The opposite occurs, for example, in Adeel et al. (2014) and Winston et al. (2014).

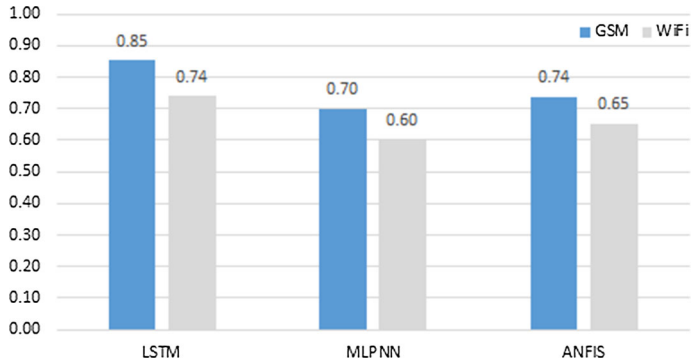

Fig. 13 Correlation coefficient in the training stage (modeling)

6 Conclusions and future work

Given the research results, the proposal for developing PU characterization algorithms that operate on input data using neural networks López et al. (2016) based on Deep Learning (as is the case for LSTM) should be considered an original contribution and a real and valid option in the search for new methodologies to minimize modeling and prediction error in the estimation of spectrum band use by primary users, thus improving performance in the spectrum decision stage for cognitive radio wireless networks. This statement is supported by the validation of results obtained with LSTM in contrast with other neural network techniques such as MLPNN and ANFIS.

In reference to the test cases in Sect. 4.2, where various PU behaviors were simulated, it was found that LSTM can easily adapt to multiple variations in traffic patterns, with a forecast accuracy above 79%; when the historical sequence has a prolonged absence/presence characteristic (as in television signals), it's possible to find estimations above 82%.

A relevant aspect of this research (in contrast with most state of the art proposals) is that the operation of the LSTM characterization algorithm was tested with real traffic sources (GSM and WiFi) and compared with MLPNN and ANFIS learning techniques. Prediction percentages achieved were in the 63.82–97.09% range. The values are higher than those produced by MLPNN (51.86– 95.59%) and ANFIS (55.86–96.30%), which proves the implementation of LSTM in real wireless systems offers promise.

Based on multiple existing characterization proposals, there is no doubt that the application of LSTM neural networks is an innovative concept for the solution of the modeling and prediction problem. This line of research should continue to be evaluated when, for example, the neural system input data sequence doesn't exhibit a binary behavior, but a continuous behavior; the response with methodologies such as ANFIS Grid or ANFIS Fuzzy C-Means Clustering, Sarima, and others should also be validated.

References

Adeel A, Larijani H, Ahmadinia. A (2014) Performance analysis of artificial neural network-based learning schemes for cognitive radio systems in LTE-UL. In: 28th International conference on advanced information networking and applications workshops (WAINA), May 13–16, Victoria, Canada, May 13–16

Akyildiz I, Lee W-Y, Chowdhury K (2009) CRAHNS: cognitive radio ad hoc networks. Elsevier, Ad Hoc Networks , vol 7, pp. 810–836. https://doi.org/10.1109/mnet.2009.5191140

Akyildiz I, Lee W-Y Chowdhury, K (2014) Spectrum management in cognitive radio Ad Hoc networks. IEEE Network 2009, vol 23, pp. 6–12. Available online: http://ieeexplore.ieee.org (Accessed on 15 Oct 2014)

Artiemjew P, Jiao N (2015) Data mining and machine learning. In: Yao Y, Hu Q, Yu H, Grzymala-Busse J (eds) Rough sets fussy sets in data mining and granular computing. Springer, Tianjin, pp 267–280

Banaei A, Georghiades C (2009) Throughput analysis of a randomized sensing scheme in cell-based Ad-hoc cognitive networks. In: IEEE international conference on communications, Dresden, Germany, June 14–18

Bkassiny M, Li Y, Jayaweera S (2011) A survey on machine-learning techniques in cognitive radio department of electrical and computer engineering. University of New Mexico, Mexico

Canberk B, Akyildiz I, Oktug S (2011) Primary user activity modeling using first-difference filter clustering and correlation in cognitive radio networks. IEEE/ACM Trans Netw 19:170–183. https://doi.org/10.1109/TNET.2010.2065031

Federal Communications Commission (2003) Notice of proposed rulemaking and order, Mexico D.F: Report ET Docket No: 03–332

Fortuna C, Mohorcic M (2009) Trends in the development of communication networks: cognitive networks. J Comput Netw 53:1354–1376

Gers F, Schmidhuber E (2001) LSTM recurrent networks learn simple context-free and context-sensitive languages. IEEE Trans Neural Netw 12:1333–1340. https://doi.org/10.1109/72.963769

Ghosh C, Pagadarai S, Agrawal D, Wyglinski A (2010) A framework for statistical wireless spectrum occupancy modelling. IEEE Trans Wireless Commun 1:38–44. https://doi.org/10.1109/TWC.2010.01.081701

Graves A (2012) Supervised sequence labelling with recurrent neural networks. Springer, Heidelberg, pp. 37–93. ISBN 978-3-642-24796-5

Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural network. In: IEEE international conference on acoustics speech and signal processing, Vancouver, Canada, May 23–26

Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In: IEEE international joint conference on neural network, Montreal, Canada, June 31–August 4

Gutiérrez L, Zazo S, Murillo J, Álvarez I, Garcia A, Pérez B (2013) HF spectrum activity prediction model based on HMM for cognitive radio applications. Elsevier, Phys Commun, vol 9, pp. 199–211. https://doi.org/10.1016/j.phycom.2012.09.004

He A, Bae K, Newman T, Gaeddert J, Kim K, Menon R, Tirado L, Neel J, Zhao Y, Reed J, Tranter W (2010) A survey of artificial intelligence for cognitive radios. IEEE Trans Veh Technol 59:1578–1592. https://doi.org/10.1109/tvt.2010.2043968

Hernández C, Salgado L, López H, Rodríguez-Colina E (2015) Multivariable algorithm for dynamic channel selection in cognitive radio networks. EURASIP J Wireless Commun Netw, vol 2015, Issue 1, pp. 1–17. Available online: http://jwcn.eurasipjournals.springeropen.com/articles/10.1186/s13638-015-0445-8https://doi.org/10.1186/s13638-015-0445-8

Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. J Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

IEEE Standard 1900.1 (2008) IEEE standard definitions and concepts for dynamic spectrum access: terminology relating to emerging wireless network, system functionality and spectrum management

Kalkan S (2015) Special topics in Deep Learning. Middle East Technical University, Ankara, Turquia, Available online: http://www.kovan.ceng.metu.edu.tr/~sinan/DL/

Khabazian M, Aissa S, Tadayon N (2012) Performance modeling of a two-tier primary-secondary network operated with IEEE802.11 DCF mechanism. IEEE Trans Wireless Commun, pp. 3047–3057

Khalid L, Anpalagan A (2014) Emerging cognitive radio technology: principles, challenges and opportunities. Elsevier, Computers and Electrical Engineering 2010, vol 38, pp. 358–366. Available online: http://www.sciencedirect.com/science/article/pii/S0045790609000524. https://doi.org/10.1016/j.compeleceng.2009.03.004. (Accessed on 20 Oct 2014)

Kwok T-Y, Yeung D-Y (1997) Constructive algorithms for structure learning in feedforward neural networks for regresission problems. In: IEEE transactions on neural networks, vol 8, pp. 630–645. Available online: http://ieeexplore.ieee.org/Xplore/home.jsp. https://doi.org/10.1109/72.572102. (Accessed on 21 June 2013)

Lee W-Y, Akyildiz I (2011) A spectrum decision framework for cognitive radio networks. IEEE Trans Mobile Comput 10:161–174. https://doi.org/10.1109/TMC.2010.147

López D, Ordoñez J, Trujillo E (2016) User characterization through dynamic bayesian networks in cognitive radio wireless networks. Int J Eng Technol 8:1771–1783

López D, Hernández L, Rivas E (2017) SVM and ANFIS as channel selection models for the spectrum decision stage in cognitive radio networks. Contemp Eng Sci 10:475–502

López D, Trujillo E, Gualdron O (2015) Elementos fundamentales que componen la radio cognitiva y asignación de bandas espectrales. Información tecnológica. vol 26, pp. 23–40. Available online: http://www.scielo.cl/pdf/infotec/v26n1/art04.pdf. https://doi.org/10.4067/s0718-07642015000100004. (Accessed on 10 Mar 2015)

Masonta M, Mzyece M, Ntlatlapa N (2013) Spectrum decision in cognitive radio networks: a survey. IEEE Commun Soc Commun Surv Tutor, vol 15, pp. 1088–1107. Available online: http://ieeexplore.ieee.org. https://doi.org/10.1109/surv.2012.111412.00160. (Accessed on 15 Aug 2014)

Masters T (1993) Multilayer feedforward networks. In practical neural network recipes in C++. Academic Press, San Diego, pp 77–116

Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, Song X, Ward R (2016) Deep sentence embedding using Long Short-Term Memory Networks: analysis and application to information retrieval. IEEE/ACM Trans Audio Speech Lang Process 24:694–707. https://doi.org/10.1109/TASLP.2016.2520371

Palangi H, Ward R, Deng L (2015) Distributed compressive sensing: a deep learning approach, Cornell University Library, arXiv. Available online: https://arxiv.org/abs/1508.04924 (Accessed on 17 Sept 2015)

Pattanayak S, Venkateswaran P, Nandi R (2013) Artificial intelligence based model for channel status prediction?: A new spectrum sensing technique for cognitive radio. Int J Commun Netw Syst Sci (IJCNS), Kolkata, April 4–7

Pedraza L, Hernández C, Galeano K, Rodriguez-Colina E (2016) Ocupación espectral y modelo de radio cognitiva para Bogotá. Universidad Distrital Francisco José de Caldas Publisher: UD Editorial

Popescu OA, Yao Y, Fiedler M, Popescu PA (2014) A management architecture for multimedia communication in cognitive radio networks. In: Fei Hu, Kumar Sunil (eds) Multimedia over cognitive radio networks. CRC Press, London, pp 3–25

Sahai A, Hoven N, Tandra R (2015) Some fundamental limits on cognitive radio. Department electrical engineering and computer science. University of California. Available online: http://www.eecs.berkeley.edu/~sahai/Papers/cognitive_radio_preliminary.pdf (Accessed on 17 Jan 2015)

Saleem Y, Rehmani M (2014) Primary radio user activity models for cognitive radio networks: a survey. J Netw Comput Appl, vol 43, pp. 1–16. Available online: http://www.sciencedirect.com/science/article/pii/S1084804514000848https://doi.org/10.1016/j.jnca.2014.04.001

Salgado L (2014) Algoritmo multivariable para la selección dinámica del canal de backup en redes de radio cognitiva basado en el método Fuzzy Analitical Hierarchical Process. Faculty of Engineering University Distrital Francisco Jose de Caldas, Bogotá

Shared Spectrum Company (2015) Spectrum reports: spectrum occupancy measurement. General survey of radio frequency bands (30 MHz to 3 GHz): Vienna, Virginia. Available online: http://www.sharedspectrum.com/papers/spectrum-reports/ (Accessed on 7 June 2015)

Sun B, Feng H, Chen K, Zhu X (2016) A deep learning framework of quantized compressed sensing for wireless neural recording. J IEEE Access 4:5169–5178

Sundermeyer M, Ney H, Schlüter R (2016) From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans Audio Speech Lang Process 23:517–529. https://doi.org/10.1109/TASLP.2015.2400218

Tumuluru V, Wang P, Niyato D (2010) A neural network based spectrum prediction scheme for cognitive radio. IEEE International Conference on Communications (ICC), Cape Town, May pp. 23–27

Uyanik G, Canberk B, Oktug S (2012) Predictive spectrum decision mechanisms in Cognitive Radio Networks. In: IEEE Globecom Workshop, Anaheim, December 3–7

Veeriah V Zhuang N, Qi G-J (2015) Differential recurrent neural networks for action recognition. In: IEEE international conference on computer vision (ICCV), Santiago, Chile, December 7–13

Wang J, Ghosh M, Challapali K (2011) Emerging cognitive radio applications: a survey. IEEE Commun Mag 49:74–81. https://doi.org/10.1109/mcom.2011.5723803

Winston O, Thomas A, OkelloOdongo W (2014) Optimizing neural network for TV Idle channel prediction in cognitive radio using particle swarm optimization. In: Fifth international conference on computational intelligence, communication systems and networks (CICSyN), Madrid, Spain, June 5–7

Yarkan S, Arslan H (2013) Binary time series approach to spectrum prediction for cognitive radio. In: IEEE vehicular technology conference, Baltimore, September/October, 30–3