




Multi-Step-Ahead Spectrum Prediction for Cognitive Radio in Fading Scenarios

Felipe G. M. Elias¹ , Evelio M. G. Fernández¹ , Vitalio A. Reguera² 

¹Department of Electrical Engineering, Federal University of Parana (UFPR), Curitiba, 81531-990, Brazil. felipe.elias@ufpr.br, evelio@ufpr.br

²Electrical Engineering Graduate Program, Federal University of Santa Maria (UFSM), Santa Maria 97105-900, Brazil. vitalio.reguera@ufsm.br

Abstract— This paper analyzes multi-step-ahead spectrum prediction for Cognitive Radio (CR) systems using several future states. A slot-based scenario is used, and prediction is based on the Support Vector Machine (SVM) algorithm. The aim is to determine whether multi-step-ahead spectrum prediction has gains in terms of reduced channel-switching and increased network throughput compared with short-term prediction. The system model is simulated in software using an exponential on-off distribution for primary-user traffic. A classical energy detector is used to perform sensing. With the help of simplifications, we present new closed-form expressions for the detection probability under AWGN and Rayleigh fading channels which allows the appropriate number of samples for these scenarios to be found. The performance of the proposed predictor is thoroughly assessed in these scenarios. The SVM algorithm had low prediction error rates, and multi-step-ahead idle-channel scheduling resulted in a reduction in channel switching by the SU of up to 51%. An increase in throughput of approximately 4% was observed for multi-step-ahead prediction with three future states. The results also show channel-switching savings can be achieved in a CR network with the proposed approach.

Index Terms— Spectral vacancies; spectrum sharing; cognitive radio.

I. INTRODUCTION

Since the publication of the seminal work by Mitola [1], Cognitive Radio (CR) has become an effective technology for solving the challenges generated by the growing demand for wireless communications. The concept of CR has already been implemented in the IEEE 802.22 [2] standard but can be further refined to improve the spectrum and energy efficiency of wireless devices. A device with cognitive capacity is called a secondary user (SU) and can coexist with primary users (PUs), which are native devices on a certain frequency band. An SU is able to check opportunistically if a channel is idle and, if so, occupy it [3].

Cognitive devices can perform spectrum sharing with a PU in three ways: underlay, overlay and interweave [4]. In the first approach, underlay, the SU can transmit simultaneously with the PU; however, the cognitive device must use mechanisms to prevent it interfering with the PU's communication so that it is effectively transparent to the PU. In the overlay method the PU

implements a mechanism involving specific keywords and techniques that report the current channel state to the SU, which is provided with data to enable it to perform interference-free transmission. In the interweave approach the SU has no prior knowledge of the PU signal or channel state and implements a mechanism for listening to the current channel state called spectrum sensing. This enables it to transmit if the PU status is idle.

Spectrum sensing can be performed in different ways, including matched filter detection, cyclostationary feature detection and energy detection [5]. The last of these is the most widely used technique because it does not require knowledge of the transmitted signal's properties, the channel state or even the modulation scheme used. The technique was first described in [6], in which a simple mechanism for detecting radio-signal energy is discussed. Two important metrics used in energy-detector design for spectrum sensing are detection probability and false-alarm probability [7], [8]. Closed expressions for these two metrics in fading channels and low-SNR scenarios can be found in the literature [9].

Using an energy-detection scheme, Liang *et al* [8] developed models with optimal sensing duration to increase data throughput for the SU network and with a high detection probability and low false-alarm probability. The higher the detection probability, the more protected the primary users will be. However, from the perspective of the SUs, the lower the probability of false alarm, the more likely the channel is to be reused when it is available. In [8], the authors describe the operation of an SU with only two stages: sensing an effective transmission of information. The SU randomly chooses one of the available channels for sensing.

Although it is possible for an SU to search randomly for idle channels, this is costly in terms of energy and search time. To ensure more successful sensing results, PU occupancy patterns can be analyzed and the future states of a channel predicted. The SU listens to the states of the channels and, after processing historical data, predicts future channel occupancy [10]. This procedure is called spectrum prediction. As the SU now knows which channels are most likely to be busy, it can selectively perform spectrum sensing. This is done first on channels that are most likely to be idle. Then, using a mechanism that allows coordination between SUs [11], the appropriate channel is chosen. Spectrum prediction yields savings in sensing energy for SUs [12] and a significant increase in the throughput of the CR network [13].

In [14] Ding *et al* study the fundamental limits of prediction of real scenarios, showing that it is valid to use pattern analysis schemes to predict the use of spectrum. Different prediction algorithms have been extensively used in the context of cognitive radio as it can be synthesized in surveys [15] and [16]. For instance, the work of Chen and Yang [17] analyzes correlations in HF electromagnetic environment in both time and frequency and developed an algorithm to predict spectrum usage in CR scenarios.

Supervised Machine Learning (ML) algorithms are of interest for spectrum prediction in CR as they can perform occupancy inference from the collected traffic data in real time. SUs can use this type of

algorithm to predict spectrum occupancy in a slot-based wireless communication system. The sequence of past slots is a data set called the observed window, and the sequence of data predicted for the future is called prediction window.

Studies on spectrum prediction using ML algorithms include the work developed by Tumuluru *et al* [18], in which the use of neural networks for spectrum prediction in CR networks is reported for the first time. Using a model known as the Multilayer Perceptron (MLP) with specific layers and short-term prediction, the authors achieved better spectrum usage and savings in sensing energy. The use of neural networks in CR is also addressed in other works, such as [12], [19], [20] and [21]. In [12], Shahid *et al* adopted an MLP neural network, while in [19], Tang and Li used a hidden multiple layer neural network.

Other types of ML algorithms described in the literature that can be used for spectrum prediction include the Hidden Markov Model (HMM) [22], nearest neighbor [22], classification trees [23], maximum likelihood [24] and Bayesian estimators. In [25], Chen *et al* developed the Minimum Bayesian Risk based Robust Spectrum Prediction algorithm which outperforms Neural-Networks prediction. Finally, in [21], Agarwal *et al* study two variations of an ML algorithm known as Support Vector Machines (SVM) - SVM with a linear kernel and SVM with a Gaussian kernel – and found that the former yielded the better predictions. The SVM algorithm was also explored in [40] where this ML classifier technique is used successfully for the detection of the QPSK signals in CR systems.

According to the amount of estimated future slots, two classes are defined, named short-term or long-term prediction. While it may seem non-trivial to define the correct amount of future slots that is encompassed by the expressions short-term prediction and long-term prediction. Sapankevych and Sankar [26] define short-term prediction as one-step ahead prediction while long-term prediction refers to the case where the prediction window consists of multiple slots [15]. In our work the term short-term prediction is associated to a single-slot prediction window. For several slots ahead, we prefer to use the expression "multi-step-ahead" since "long-term" would be better applied to cases with much larger slot amounts than those studied in this paper. Multi-step-ahead slot prediction is in the early stages of development with enormous challenges ahead [15] due the error accumulation problem. Multi-step-ahead prediction is reported in [27], where Min *et al* describe an algorithm based on multiple slots prediction is proposed to forecast the interference time ratio culminating to avoid interference in the opportunistic spectrum access (OSA). In [12] only multiple idle slots are accounted for prediction using a Neural Network.

Approaches using Neural Network for multiple slots analysis were also found in references [19] and [28]. In [19] Tang and Li analyses Deep Recurrent Neural Network with several hidden layers is used to learn spectrum occupancy for WiFi channels. In [28] Shawel *et al* employ Convolutional Long Short-Term Memory (ConvLSTM) Deep Learning Neural Network is employed for long-term prediction to learn joint spatial-spectral-temporal dependencies in spectrum utilization. Multi-step-ahead prediction efforts are also presented by Sun *et al* [29] and by Ge *et al* [30], where both works

employ time frequency-day spectrum tensor model used for long-term prediction. The proposal in [30] outperforms mechanisms presented in [29], fixing the problem of missing data and sparse anomalies.

Bayhan and Alagoz [31] and Chang *et al* [32], discuss the channel-switching cost associated with CR, as a consequence of the need for radio-circuit reconfiguration when switching channels.

Short-term prediction in conjunction with channel-sensing is described in [13], which reported gains in terms of throughput using MLP prediction. In [13], the prediction stage in the framework proposed by [8] was added prior to the sensing stage to allow the SU to perform sensing only on channels predicted as idle. Both studies [8] and [13], considered only an additive Gaussian white noise (AWGN) channel and did not analyze a fading channel.

To the best of the authors' knowledge, joint analysis of multi-step-ahead prediction and sensing in fading environments was not reported in literature. In addition, multi-step-ahead prediction study with SMV was still not validated. The main contributions of this work are:

- The multi-step-ahead joint prediction sensing analysis for the purpose of resourceful channel usage in CR network. This scheme allows channel-scheduling to be implemented as the SU can choose the channel with the largest number of consecutive time slots predicted as idle allowing for a lower number of channels switching. The prediction is realized with SVM ML, attaining a low probability of prediction error. Expressions for the probability of a cognitive radio network with multiple channels and multi-step-ahead prediction are theoretically derived.
- A new SU frame with multi-step-ahead prediction and sensing is defined and it is determined whether there is a resulting increase in throughput in the CR network. For this research and future studies, the handoff (channel-switching) is very important and must be analyzed for a more realistic point-of-view of SU frame. The handoff affects the entire framework of the SU, as the time spent by the SU in this process decreases the transmission and the production of results consistent with the real scenarios. For this reason, this work introduces an algorithm that tries to prevent as much as possible the handoff of channels.
- A simplified expression for determining the detection probability of the energy detector for Rayleigh fading channels is proposed which allows the number of required samples for reliable detection in fading scenarios to be accurately determined. This is motivated because Rayleigh fading channel severely attenuates the signal and impacts primarily energy detection other than spectrum prediction. Consequently, a low probability of detection influences the data feeding the predictor algorithm, resulting in a severe reduction of predictor performance.

The results show that there was an improvement in the energy-detector performance in Rayleigh fading scenarios. There was a reduction in channel switching and an increase in CR network throughput when multi-step-ahead prediction schemes were used for both an AWGN channel and a Rayleigh fading scenario. The rest of this work is divided as follows Section II describes the system

model used here. Section III shows the proposed improvements. Section IV presents the results of the simulation. Finally, in section V conclusions are drawn.

II. SYSTEM MODEL

Figure 1 shows a CR network consisting of a SU transmitter and a SU receiver which coexist with native PU users. This Figure also shows a primary base station that establishes communication with several PU equipment, each link is established using a single channel. The PU equipment can access N_C licensed channels, which can also be accessed opportunistically by the SU equipment. In Fig. 1, the SU device is sensing all the N_C licensed channels. When a channel is available, the SU can access it to transmit to the corresponding cognitive receiver. The SU receiver is aware of the channel in use by the SU transmitter by means of a Common Control Channel (CCC) with limited bandwidth. Interference between an SU and PU can occur if spectrum-detection errors occur, in which case the SU device transmits based on the erroneous assumption that the analyzed channel is idle. The occupancy of any given channel due to PU activity is assumed to be independent of occupancy in any other channel. One PU can occupy only one channel at a time and the channels are assumed to be subject to Rayleigh fading.

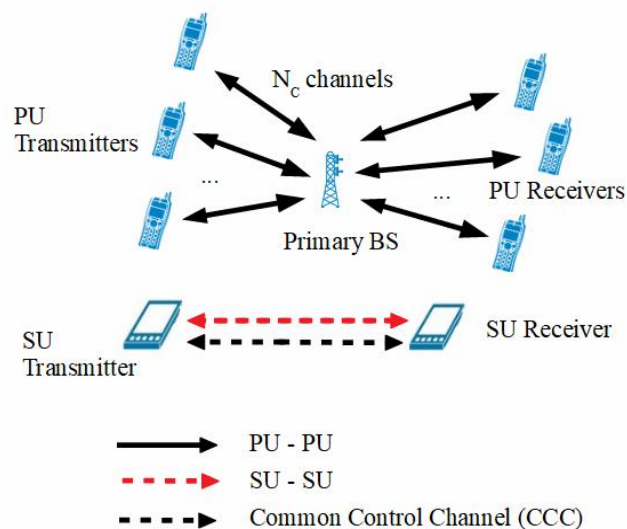


Fig. 1. Proposed system model.

When designing a wireless system, the Rayleigh fading model is one of the most appropriate to consider, as it serves the purpose of modeling a transmission without line of sight, such as dense urban areas [7].

A stochastic process with two states models PU traffic occupancy: busy or idle. Simulations were performed with an exponential on-off traffic distribution model, in which the interval when the

channel is busy follows an exponential distribution with mean μ_1 and the interval when it is idle has mean μ_0 . The system model is composed of multiple channels, each one occupied by one PU.

The history of PU occupancy is modeled by binary states denoted $s_i \in \{0,1\}$, where s_i is the state of the slot being analyzed. A busy slot is represented by 1, and an idle slot by 0.

III. PROPOSED IMPROVEMENTS

In the works referred to in the previous sections the slot or frame is divided into stages. In [8] only two stages are presented: sensing and transmission; in [13], in contrast, the frame is divided into three stages: prediction, sensing and transmission. In order to deal with multi-step-ahead prediction we made two modifications to the SU frame structure, these can be seen in Fig. 2, which shows a series of larger frames, made up of an initial frame and $F - 1$ subsequent frames, where F is the number of predicted future states. The first frame has a duration T and is divided into four stages: prediction, with a duration τ_P ; sensing, with a duration τ_S ; channel-switching latency or delay, with a duration τ_{SC} ; and the effective transmission of information, lasting $T - \tau_P - \tau_S - \tau_{SC}$. The next $F - 1$ frames are segmented into up to three stages in a similar manner: sensing, channel-switching latency and transmission. In these frames the actual transmission has a duration $T - \tau_S - \tau_{SC}$. After the $F - 1$ three-stage frames, the SU begins the prediction cycle again with the initial frame divided into four stages.

The prediction stage, which occurs only in the initial frame, is performed with an ML algorithm. In this stage, the SU estimates the state of N_C channels based on historical data and selects only those that are considered idle to perform sensing. In the second stage (sensing), a channel that has been predicted to be idle is chosen at random and the energy-detection technique is used to evaluate the actual state of the channel. Data transmission takes place if the channel is in fact idle. Channel switching occurs if the SU determines that the current channel will be busy [31], [32]. Channel switching results in latency because RF (radio frequency) circuitry must be reconfigured. If the predicted channel is the same as that already occupied by the SU, this stage is not counted, increasing the effective information transmission time. This model is more realistic than previous schemes because it considers the channel-switching latency and does not repeat the prediction stages.

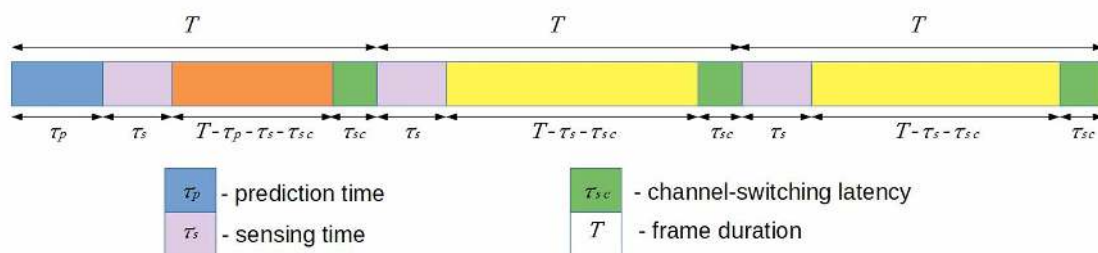


Fig. 2. SU slot or operating frame with multi-step-ahead prediction, sensing and channel switching. The effective transmission time is denoted either as $T - \tau_P - \tau_S - \tau_{SC}$ or $T - \tau_S - \tau_{SC}$.

The channel-switching latency (handoff) affects the entire SU frame, the time spent with handoff impacts the effective transmission decreasing the throughput and it is relevant for the analysis. Its consideration is very important as this work presents an algorithm that tries to avoid channel switching as much as possible.

A. SVM and Prediction Stage

Prediction in the context of CR can be performed through different techniques described in the literature including Machine Learning (ML) algorithms. ML algorithms can be divided into two categories, classifiers and regressors. Classifiers perform the classification by estimating discrete values. One example is the Support Vector Machine (SVM) algorithm, this was developed in [33] and it is used in many areas of science such as market prediction, energy load estimation, among others.

Like other ML techniques, standard SVM performs data classification, it is considered a non-probabilistic binary linear classifier. It takes a training stage with a set of samples taken from the total space problem data. The algorithm performs a process of categorizing the input data into just one of two classes. One way of viewing the operation of the algorithm is to treat each input data as a point in a hyperplane. The algorithm then finds a line with the maximum distance between two types of data categories. After executing the algorithm, the output is associated with only one of these categories. The purpose of the algorithm is to maximize the differences between the two categories to achieve correct classification.

The main reasons for choosing the SVM technique instead of the MLP neural network as a prediction tool are the small number of free parameters and the guarantee of converging to the optimal solution [26]. Although the training phase of the two tools is computationally expensive, at the time of execution with the real data set, both have the advantage of being very efficient.

Given a data set $x(t)$, where t is a series of discrete samples $t = \{0, 1, 2, 3, 4, 5, N - 1\}$, an estimated value in the future is defined as $y(t + \delta)$. The classification aims at defining the following equation:

$$f(x) = w \cdot \phi(x) + \varepsilon, \quad (1)$$

where w is the weight of the prediction function and ε is called limit value. The function $\phi(x)$ is the kernel function which is used when the input data space is not linear. The kernel function is a procedure to map the $x(t)$ data to a higher dimension and then to perform linear regression. Several kernel functions meet the necessary conditions for mapping, such as hyperbolic, Gaussian and polynomial tangents [26]. In the training phase, the objective is to find the optimal weights and limit values. One of the criteria is to find values for w with similar levels between them, this is verified by the Euclidean norm. Another criterion performed in the training phase is to reduce the error generated

in the value estimation process; the residual error is given by:

$$R_{emp}(f) = \frac{1}{N} \sum_{i=0}^{N-1} L(x(i), y(i), f(x(i), w)), \quad (2)$$

where $L(\cdot)$ is a cost function which depends on the input $x(t)$, prediction $y(t)$ and regression function (1). Lagrange multipliers are used to solve this kind of convex optimization problem, which can be executed in linear arrangement in consequence of the kernel mapping process.

In this work, we used the linear kernel instead of the Gaussian kernel. Empirical tests showed a higher performance for the linear kernel which is aligned with the work of [21] that presented superior performance of the linear kernel.

For a given channel $\eta \in N_C$, the observed window (past-data window) has length L and is defined as $s_{\eta}^L = \{s_{\eta}^{i-1}, s_{\eta}^{i-2}, s_{\eta}^{i-3}, \dots, s_{\eta}^{i-L}\}$. The prediction window (future-data window) has length F and is defined as $\hat{s}_{\eta}^F = \{\hat{s}_{\eta}^{i+1}, \hat{s}_{\eta}^{i+2}, \hat{s}_{\eta}^{i+3}, \dots, \hat{s}_{\eta}^{i+F}\}$. Real future data are defined as $s_{\eta}^F = \{s_{\eta}^{i+1}, s_{\eta}^{i+2}, s_{\eta}^{i+3}, \dots, s_{\eta}^{i+F}\}$. Short-term prediction is the simplest prediction scheme, where $F = 1$. In this case, $\hat{s}_{\eta}^F = \hat{s}_{\eta}^{i+1}$ and $s_{\eta}^F = s_{\eta}^{i+1}$. For multi-step-ahead prediction, $F > 1$.

The training phase consists of providing the algorithm with a large amount of data, previously known and in this case, separated in several observed windows and its associated future-data windows. The algorithm in its training phase acts to minimize the error between the values of the predicted future window and the already known future-data window.

The MATLAB 2018 software was used to generate the simulation environment. This work utilizes the multiclass model for Support Vector Machine [35] presented in MATLAB native Statistics and Machine Learning Toolbox. As mentioned before, we adopted the linear kernel parameter and the SVM algorithm performs the classification for multiple classes. Although we use binary data to define spectrum occupation or idle states, in our implementation we chose to leave the observed window in binary state and we chose to perform a simple operation in the future-data window by aggregating the binary values within a future window with decimal value by performing a simple binary-decimal conversion. This simple conversion takes place before entering the data in the predictor algorithm, and this scheme is first used in the training phase. For example, considering $F = 3$, the algorithm defines $2^F = 8$ possible different classes, a direct result of the binary combination of the three predicted slots. In its training phase, therefore, the algorithm makes the association between an observed window (in binary format) with a decimal number. In the validation phase when the predictor algorithm receives an observed window, it returns a decimal value (multiclass). After performing the prediction by the SVM algorithm in the validation phase of the real data in a multiclass format, we perform the inverse decimal-binary conversion. As stated, the input data (observed window) are not aggregated into multiple classes but are fed into the predictor as binary data.

The predicted vector \hat{s}_η^F can also be analyzed in terms of the number of future idle slots. This analysis is important for the scheduling algorithm, which will be described in more detail in Section III-D. The number of future idle slots is counted from the current slot and only takes into account sequential idle slots. The SU can use this information to choose a channel that has been predicted to be idle the longest. The number of subsequent sequential idle slots is denoted by \hat{u} where $\hat{u} \leq F$ and $\hat{s}_\eta^u = \{s_\eta^{i+1}, s_\eta^{i+2}, s_\eta^{i+3}, \dots, s_\eta^{i+u}\}$. As an example, if a multi-step-ahead future window with $F = 5$ equals $\hat{s}_\eta^F = \{0, 0, 0, 1, 0\}$, the vector that considers only the next idle sequential slots is given by $\hat{s}_\eta^u = \{0, 0, 0\}$. This number can also be represented in decimal format; hence, in this example $\hat{u} = 3$, *i.e.*, there are only three sequential future idle slots.

In this work, imperfect prediction was considered because the SVM technique can make erroneous predictions, causing the predicted value for slot n , \hat{s}^{i+n} , to be different from its real value, s^{i+n} . The probability of prediction error is denoted P_p^e .

B. Sensing Stage

Channels that are predicted to be idle are selected for the second stage of the SU frame, the sensing stage. Energy detection [6] is a well-known sensing technique in which the output is proportional to the energy of the received signal. The amount of energy detected is compared with a threshold, and a state corresponding to either the presence or absence of a signal at the receiver is returned. The result can be presented as a binary hypothesis on occupancy:

$$\begin{cases} \mathbf{H}_0: y(t) = n(t) \\ \mathbf{H}_1: y(t) = hx(t) + n(t), \end{cases} \quad (3)$$

Noise is represented by $n(t)$, h is the channel gain, $x(t)$ is the PU signal, and $y(t)$ is the signal received at the SU. Under hypothesis \mathbf{H}_0 only noise is present at SU receiver, while under hypothesis \mathbf{H}_1 the PU signal is also present at the SU receiver side. The channel-occupancy probabilities, which are shown in Section II for both hypotheses, are functions of traffic density. We also assume slow or quasi-static fading such that the channel gain h remains constant over several time-slots.

Noise samples $n(t)$ are assumed to be AWGN with zero mean and variance σ_n^2 . The signal-to-noise ratio (SNR) between the PU and SU is given by $\gamma = h \sigma_x^2 / \sigma_n^2$, where σ_x^2 is the average power of $x(t)$. The received signal $y(t)$ passes through the energy detector, and the output of the detector is used to determine the test statistic, E . The detection probability P_d and the probability of false alarm P_f are set according to the test statistic as

$$\begin{cases} P_d = P[E > \lambda | \mathbf{H}_1] \\ P_f = P[E > \lambda | \mathbf{H}_0], \end{cases} \quad (4)$$

where λ is the detection threshold.

Closed-form equations for P_d and P_f in AWGN channels have already been derived in [8], [34], as well as the approximations

$$P_d \approx Q\left(\frac{\lambda - N(2\sigma_n^2)(1 + \gamma)}{\sqrt{N(1 + 2\gamma)}(2\sigma_n^2)}\right) \quad (5)$$

$$P_f \approx Q\left(\frac{\lambda - N(2\sigma_n^2)}{\sqrt{N}(2\sigma_n^2)}\right). \quad (6)$$

Where N represents the number of samples and $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du$ is the Gaussian Q -function. One of the advantages of working with (5) and (6) is that not only are they simpler forms, but the inverse Q -function, Q^{-1} , can be used. Thus, by rearranging (6), we can obtain the minimum value of λ for a given design value \widehat{P}_f , as follows:

$$\lambda^* = (Q^{-1}(\widehat{P}_f) + \sqrt{N}) \sqrt{N} 2\sigma_n^2 \quad (7)$$

In scenarios with low SNR, *i.e.*, low values of γ , it is more difficult to detect a signal effectively and the energy detector therefore needs more samples.

According to [8], the minimum number of samples, N , required for effective detection in an AWGN channel is:

$$N = [Q^{-1}(\widehat{P}_f) - Q^{-1}(P_d)\sqrt{2\gamma + 1}]^2 \gamma^{-2} \quad (8)$$

Equation (8) shows that for the AWGN channel model the number of samples depends only on the design values of \widehat{P}_d , \widehat{P}_f and γ . In the case of the AWGN channel the number of samples is denoted by N while in the case of the Rayleigh channel it is referred to as N_{Ray} . Signal detection should be reliable even in a low-noise environment and comply with design or standard constraints. The IEEE 802.22 WRAN standard sets $\widehat{P}_f = 0.1$ even under conditions where the signal-to-noise ratio is extremely low, *e.g.*, $= -20dB$, and P_d is defined as a design performance parameter. In the case of a low SNR constraint, the signal has little impact on the variance of the statistic under hypothesis \mathbf{H}_1 . As $\gamma \ll 1$, Equation (5) can be simplified by canceling the term γ in the denominator:

$$P_d = Q\left(\frac{\lambda - N(2\sigma_n^2)(1 + \gamma)}{\sqrt{N}(2\sigma_n^2)}\right) \quad (9)$$

Knowing that $Q(x) = 1/2 - 1/2 \operatorname{erf}(x/\sqrt{2})$ and the approximation $\operatorname{erf}(x) = [2/(1 + e^{-2.5x})] - 1$, as a simplification of the results found in [36], Equation (9) can be rewritten as:

$$Q(x) = \frac{1}{2} - \frac{1}{2} \left(\frac{2}{1 + e^{-\frac{2.5x}{\sqrt{2}}}} - 1 \right) \quad (10)$$

Combining (5) and (10) gives the unprecedented closed-form expression for P_d at low-SNR regime:

$$P_d = \frac{1}{2} + \frac{1}{2} \left(1 + \frac{2}{1 + e^{-\left(\frac{5(\lambda - N(2\sigma_n^2)(1+\gamma))}{4\sqrt{2}N\sigma_n^2} \right)}} \right) \quad (11)$$

This equation is an excellent approximation for (9), as can be seen in section IV-B, and greatly simplifies the algebraic manipulation required, as can be seen below.

C. Rayleigh Channel Energy Detector

The instantaneous detection probability P_d depends on the SNR and the number of samples as can be seen in (5), and may also be affected by the state of the wireless channel. Shadowing and fading effects on the channel, for example, make it necessary to evaluate the average detection probability \overline{P}_d which can be obtained by averaging P_d over the distribution of the channel SNR [34], $f(\gamma)$, which gives:

$$\overline{P}_d = \int_0^{\infty} P_d(\gamma) f(\gamma) d\gamma \quad (12)$$

In a Rayleigh fading scenario, the distribution of the SNR is $f_{Ray}(\gamma) = \frac{1}{\gamma} \exp(-\frac{\gamma}{\bar{\gamma}})$. Using $f_{Ray}(\gamma)$ and (11) in (12), and with some algebraic manipulation, gives the novel expression

$$\overline{P}_d = {}_2F_1 \left[1, \frac{2\sqrt{2}}{5\gamma\sqrt{N}}, 1 + \frac{2\sqrt{2}}{5\gamma\sqrt{N}}, -\exp\left(\frac{5Q^{-1}(P_f)}{2\sqrt{2}}\right) \right]. \quad (13)$$

In the above expression, ${}_2F_1$ is a Gauss hypergeometric function of the form ${}_2F_1(a, b, c, z)$ [37]. This equation is a compact form for $\overline{P}_{d(Ray)}$ found in [34] and allows the specific number of samples N_{Ray} for a given value of $P_{d(Ray)}$ to be determined by iteration.

D. Transmission Stage

The last stage of the SU frame is the effective transmission of information based on the prediction and sensing stages. We will present the same calculations developed in [13] for $F = 1$, next we will extend the calculations for $F > 1$.

When prediction is made for only one future state, *i.e.*, $F = 1$, and considering only one channel, let $P_{0(s)}$ and $P_{1(s)}$ denote, respectively, the probabilities of the prediction stage classifying this channel as idle or busy which are given by

$$P_{0(s)} = P(H_0)(1 - P_p^e) + P(H_1)(P_p^e) \quad (14)$$

$$P_{1(s)} = P(H_0)P_p^e + P(H_1)(1 - P_p^e). \quad (15)$$

According to [13], considering the N_C channels, the prediction probability indicating idle channel availability is

$$P_{0(s|N_C)} = \sum_{k=1}^{N_C} \binom{N_C}{k} (P_{0(s)})^k (P_{1(s)})^{(N_C-k)}, \quad (16)$$

and the probability of all the channels being predicted to be busy is

$$P_{1(s|N_C)} = \binom{N_C}{0} (P_{0(s)})^0 (P_{1(s)})^{(N_C-0)} = (P_{1(s)})^{N_C}. \quad (17)$$

When this last event happens, the SU selects a random channel to sense. At no time do the prediction results affect the sensing result.

The joint probability distribution of events prediction, sensing and the true state for only one channel $(P_{0[j|1]}, P_{1[j|1]}, \dots, P_{7[j|1]})$ and for the entire network $(P_{0[j|N_C]}, P_{1[j|N_C]}, \dots, P_{7[j|N_C]})$ considering prediction for only one future state ($F = 1$) was derived in [13] and is shown in Table III of [13]. The first subscript number in the probabilities is the decimal representation of the binary sequence representing the $2^3 = 8$ different joint events.

Extending the analysis to multiple future states ($F > 1$), for the sake of simplicity we first consider $F = 2$, *i.e.*, the SU algorithm analyzes only two future states, and then generalize the results to any F . For only one channel, the probability of predicting both states as idle is $P_{0(l)} = (P_{0(s)})^2$. The probability of predicting the first state to be idle and the next to be busy is $P_{1(l)} = (P_{0(s)})(P_{1(s)})$. Similarly, the probability of predicting the first state to be busy and the next to be idle is $P_{2(l)} = (P_{1(s)})(P_{0(s)})$. Finally, the probability of predicting both states to be busy is $P_{3(l)} = (P_{1(s)})^2$.

Note that there are $B = 2^F$ different combinations of idle-busy prediction results when considering F future slots. Let $b \in \{0, 1, \dots, (B - 1)\}$ denotes the decimal representation of a combination of idle-busy prediction results. For instance, considering three future slots, $b = 5$ means that the prediction result is (busy, idle, busy), which is represented by the binary sequence $\{101\}$. Then, for any F the probability of a combination of idle-busy prediction result can be computed as

$$P_{b(l)} = (P_{0(s)})^\xi (P_{1(s)})^\kappa, b = \{0, 1, \dots, (B - 1)\}, \quad (18)$$

where ξ is the number of idle predicted states and $\kappa = F - \xi$ is the number of busy predicted states.

Now, considering the N_C channels and multiple slots, the prediction probability indicating idle channel availability can be considered as a multinomial experiment. This kind of experiment consists of a number N_C of trials, since each channel can be viewed as independent of each other. Each attempt can result in any one of B possible outcomes: O_0, O_1, \dots, O_{B-1} . In the previous example considering $F = 2$, the four possible outcomes (or events) are: $O_0 = \{00\}, O_1 = \{01\}, O_2 = \{10\}, O_3 = \{11\}$. The probabilities $P_{0(l)}, P_{1(l)}, \dots, P_{B-1(l)}$, are associated to each possible outcome. Let define the weight vector $(k_0, k_1, \dots, k_{B-1})$ which specify the number of times each outcome O_b occurs, such that $k_0 + k_1 + \dots + k_{B-1} = N_C$. The probability associated with this vector is given by:

$$P[(k_0, k_1, \dots, k_{B-1})] = \binom{N_C}{k_0, k_1, \dots, k_{B-1}} \prod_{b=0}^{(B-1)} (P_{b(l)})^{k_b}, \quad (19)$$

where,

$$\binom{N_C}{k_0, k_1, \dots, k_{B-1}} = \frac{N_C!}{k_0! k_1! \dots k_{B-1}!}, \quad (20)$$

is the multinomial coefficient. Note that this is the probability that O_0 occurs k_0 times, O_1 occurs k_1 times, ..., and O_{B-1} occurs k_{B-1} times. Still using $F = 2$ we set another example, in order to facilitate understanding of the prediction technique with multiple slots. Under the perspective of multiple channels, considering $N_C = 3$, the weight vector has four elements (k_0, k_1, k_2, k_3) and a possible value for the vector could be $(2, 1, 0, 0)$ as $2 + 1 + 0 + 0 = 3$.

This means that O_0 occurs two times, O_1 occurs one time and there are no occurrences for both O_2 and O_3 . The probability of this event is given by:

$$\begin{aligned}
 P[(2,1,0,0)] &= \binom{3}{2,1,0,0} \prod_{b=0}^{(3)} (P_{b(l)})^{k_b} \\
 &= \frac{3!}{2! 1! 0! 0!} (P_{0(l)})^2 (P_{1(l)}) \\
 &= 3(P_{0(l)})^2 (P_{1(l)})
 \end{aligned} \tag{21}$$

The prediction probability indicating idle channel availability across the N_C -channel network is a union of probabilities like the one shown above.

Therefore, among all possible state combinations of vectors, we are only interested in those that involve idle-state sequences starting at \hat{s}^{i+1} . In the previous example considering $F = 2$ there are two such sequences, $O_0 = \{00\}$ and $O_1 = \{01\}$, that happen with probabilities $P_{0(l)}$ and $P_{1(l)}$, respectively.

The other two combinations (that happen with probabilities $P_{2(l)}$ and $P_{3(l)}$) start their sequences with busy state and are not evaluated by the scheduling algorithm. Another example can be visualized if we consider $F = 3$, which leads to $B = 2^3 = 8$ possible different prediction probabilities denoted as $P_{0(l)}, P_{1(l)}, \dots, P_{7(l)}$. In this last case the scheduling scheme will analyze only four predicted sequences: $\hat{s}_\eta^3 = \{000\}, \{001\}, \{010\}$ and $\{011\}$ which happen with probabilities: $P_{0(l)}, P_{1(l)}, P_{2(l)}$ and $P_{3(l)}$. Note that there can be up to $\frac{B}{2}$ possible idle-busy prediction results starting with an idle state, and consequently, can be elected to be sensed.

The elements of the vectors elected to compose the aforementioned union should satisfy $\{k_0, k_1, \dots, k_{\frac{B}{2}-1}\} \geq 1$, which means that the number of times each outcome where idle-state sequences starting at \hat{s}^{i+1} should not be zero and all the eligible cases should be added. The final probability is therefore a sum of multinomial distribution terms for different weight vectors (which respect the specified criteria).

By performing the analysis for the entire CR network made up of N_C channels, we get an extension of the result presented in [13]. The generalized equation for the probability of the entire network with multiple channels being idle in multiple future states is therefore

$$\begin{aligned}
 P_{0(l|N_C)} &= \sum_{k_0 \in \phi} \sum_{k_1 \in \phi} \dots \sum_{k_{\frac{B}{2}} \in \phi} P[(k_0, k_1, \dots, k_{B-1})] \\
 \phi &= \{k_i \neq 0 | k_0 + k_1 + \dots + k_{B-1} = N_C\}
 \end{aligned} \tag{22}$$

Equation (22) is original, and takes into account all cases where there is a sequence of idle channels starting with \hat{s}^{i+1} . The probability of the entire network being busy is given by

$$P_{1(l|N_C)} = 1 - P_{0(l|N_C)} \tag{23}$$

Which accounts for the events: all channels busy in all future states or only some channels busy in future states.

In the case of two future states, three more columns are added to Table III of [13]. As the number of future states increases, the number of rows in the table increases exponentially. Assuming the case where $F = 2$ and considering the analysis of only one channel, if the prediction is idle, the sensing is idle and the true state is also idle for all $F = 2$ future states analyzed we will have the extension of the result of [13] and the result in this particular case is the square of the first row, third column of Table III of [13]. The total quantity of possible combination for $F = 2$ and just one channel is $2^{(2 \times 3)} = 64$ rows, generalizing we have $\psi = 2^{(F \times 3)}$ different probability of true channel state, prediction and sensing with $F = 2$ and just one channel.

The theoretical probability distribution of true channel state, prediction and sensing for $F > 1$ and the entire network (multiple channels), can be found by multiplying the probabilities of the first term of equations from third column of Table III of [13] and the probability equations for the whole network, (22) and (23):

$$P_{\psi} = \left(\prod_{w=1}^F P_w^1 \right) P_{g(l|N_C)} \quad (24)$$

where $g \in 0,1$, which selects equations (22) or (23). The index ψ is described above and $\psi \in 1, \dots, 2^{(F \times 3)}$. The product inside the brackets in (24) is associated to the quantity of future states analyzed, thus we have a sequence of multiplications of the eight rows in Table III of [13] and as a result $w \in 1, \dots, 8$.

The throughput of a CR network is defined as the total amount of data transmitted divided by the total transmission time. If a channel is sensed busy, the SU will not transmit data and the throughput will be zero, *i.e.*, $C_0 = 0$, where index 0 indicates the non-transmission state. If a channel is sensed idle and is indeed idle, and if the gain of the channel, h , which is subject to Rayleigh fading, is assumed to be an ergodic process [38], [39], the throughput of the channel will be

$$C_1 = \alpha \cdot \log_2 e \cdot \exp\left(\frac{1}{\gamma_S}\right) \expint\left(\frac{1}{\gamma_S}\right). \quad (25)$$

This is the throughput in the best case because there is no interference from the PU in the chosen channel.; γ_S is the SNR between the transmitting and receiving SU, $\expint(x) \triangleq \int_x^{\infty} \frac{e^{-t}}{t} dt$ and the factor α is given by

$$\alpha = \frac{FT - F\tau_S - \tau_P - \tau_{SC}}{FT}. \quad (26)$$

As multi-step-ahead prediction estimates F states into the future, the prediction stage does not need to be run more than once, so there is a slight increase in network throughput. When the channel is sensed idle but is actually occupied by the PU, the throughput will be reduced as there will be interference from the PU in the channel. Therefore

$$C_2 = \alpha \cdot \log_2 e \cdot \exp\left(\frac{(1 + \gamma)}{\gamma_S}\right) \exp\left(-\frac{(1 + \gamma)}{\gamma_S}\right), \quad (27)$$

corresponds to the transmission capacity in the least favorable circumstances. The network throughput, R , is given by

$$R = \left(\sum_{\text{Sens.idle-Real:idle}} P_\psi \right) C_1 + \left(\sum_{\text{Sens.idle-Real:busy}} P_\psi \right) C_2. \quad (28)$$

Both summations refer to cases where sensing was performed but the actual channel state changes the associated throughput. Equation (28) is normalized to generate dimensionless values by dividing it by the upper limit $R_S = P(H_0)C_1 + P(H_1)C_2$:

$$R_{Norm} = \frac{R}{R_S} \quad (29)$$

E. Channel Scheduling

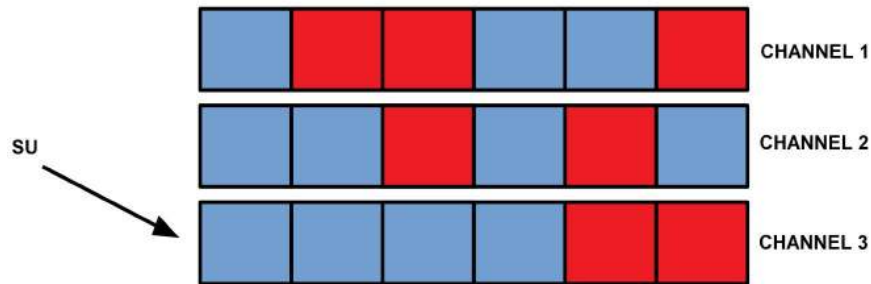
Channel scheduling increases the efficiency of a CR network by reducing the amount of channel switching, reducing transmission pauses and making communication more stable. Channel switching is investigated in [12] in the context of increased transmission quality in LTE networks. In multichannel communication using timeslots, the user should remain on the same channel for as long as possible as channel switching involves hand-off mechanisms that can result in computational costs and delays for the end user.

In this context, a change in the chosen channels for sensing compared with other studies found in the literature was developed. Because the multi-step-ahead predictor can estimate multiple slots ahead, it determines which channels will be idle longer and which will have the greatest number of sequential idle slots.

The proposed algorithm implemented in the SU looks for the channel with the greatest possible number $\hat{\eta}$ of idle future states among the N_C available channels. Thus, channels for which all idle states are prioritized for sensing. Figure 3a shows busy slots in red and idle slots in blue. As the choice of channel is based on multi-step-ahead prediction, the SU selects the third channel, as

illustrated. In the case of a traditional prediction algorithm, the SU could choose any of the three available channels as the first slot in each of them is considered idle.

The proposed algorithm also yields an improved result for $F = 1$ and gives preference to the channel that had already been chosen in the previous state, avoiding a potentially unnecessary channel change. The pseudo code for the algorithm can be verified in Algorithm 1 (depicted in Fig. 3b).



ALGORITHM 1 CHANNEL SCHEDULING

```

1: initialization;
2: i = actual slot;
3: if F > 1 then
4:   for each  $\eta$  in  $N$  do
5:      $u_\eta = 0$ ;
6:     for each j in  $F$  do
7:       if  $\hat{S}_\eta^j = 0$  then
8:         increment  $u_\eta$ ;
9:       else
10:        break;
11:      end if
12:    end for
13:  end for
14:   $\hat{\eta}^{i+1} \leftarrow \eta$  [ $\max(u_\eta)$ ];
15: else
16:  for each  $\eta$  in  $N$  do
17:     $u_\eta = 0$ ;
18:    if  $\hat{S}_\eta^{j+1} = 0$  then
19:       $u_\eta = 1$ ;
20:    end if
21:  end for
22:  if  $u_\eta = 1$  and  $S_\eta^i = 0$  then
23:     $\hat{\eta}^{j+1} \leftarrow \eta$ ;
24:  else if  $u_- = 1$  then
25:     $\hat{\eta}^{j+1} \leftarrow \eta$ ;
26:  else
27:     $\hat{\eta}^{j+1} \leftarrow \text{rand}(\eta)$ ;
28:  end if
29: end if
    
```

Fig. 3. (a) Channel selection with the most sequential future idle slots; (b) The pseudo code for the algorithm.

IV. RESULTS

In this work we used a training set with $T_r = 1000$ sample slots, and the total number of validation slots per simulation round was set to $S = 30000$, the same value as that used in [12]. Simulations were performed for training and performance evaluation. As described in Section III-A, each SU was trained with different slots to avoid biased results. For the remainder of the simulation, the same design values as those in [13] were used according to Table I.

TABLE I - THE SIMULATION PARAMETERS FOR THE PROPOSED IMPROVEMENTS

Parameter Name	Parameter variable	Value	Parameter name	Parameter variable	Value
False alarm probability	P_f	0.1	Sensing time	τ_S	2.5 ms
Detection probability	P_d	0.9	Channel switching latency	τ_{SC}	5 ms
Frame duration	T	100 ms	SU to SU SNR	γ_S	20 dB
Prediction time	τ_P	5 ms	PU to SU SNR	γ	-20 dB

Where the SNR of the PU sensed by the SU complies with the design described in Section III-C. The procedures, steps, equations and algorithms¹ presented in Sections II and III were implemented through the MATLAB® 2018 Software from Mathworks® in a personal computer (PC), where the “Statistics and Machine Learning Toolbox” and “Neural Network Toolbox” are required. The hardware of the PC is a Core i5-7300HQ CPU @2.5 GHz, 8GB DDR4@ 2400MHz. The algorithms were tested using the following MATLAB functions: “fitcknn”, “fitctree”, “fitsvm”, “fitecoc”, “predict”, “network” and “train”. It was possible to perform several computer simulations that validate the main contributions of this work and present a comparison with what was provided in the literature.

A. Imperfect Prediction

The SVM technique was chosen after a thorough comparison between several ML techniques. Four machine learning algorithms were compared, namely: k-Nearest Neighbor [22], Classification Tree[23], Neural Network [18] and SVM. Each algorithm was tested with a set of data used in this work, that is, a set of spectrum occupation slots with the possible states 1 or 0, with their traffic following the Poisson-binomial distribution.

Figure 4 shows the probability of prediction error, P_p^e , a measure of prediction performance as defined in Section III-A. This quantity is obtained by measuring the predicted data and the actual data. The figure shows the probability of prediction error as a function of the amount of future data. As expected, the level of uncertainty always increases as we increase the quantity of predicted points, this is known as the error accumulation problem [15]. In the figure it can be observed that the SVM algorithm offers the best performance among all the other tested algorithms. This feature of SVM ML

¹ – The MATLAB code can be found in this repository : <https://github.com/felipeelias-ops/SpectrumPrediction>

holds for all tested number of future slots. The k-Nearest Neighbor and Classification Tree techniques were 11% and 13% faster compared to the SVM algorithm in the training phase. The Neural Network technique was 16% slower than the SVM algorithm in the training phase. In the prediction phase, the k-Nearest Neighbor and Classification Tree algorithms were 6% and 8% faster compared to the SVM technique, while the Neural Network technique was 4% slower. The SVM was still the best option as it provides the best accuracy.

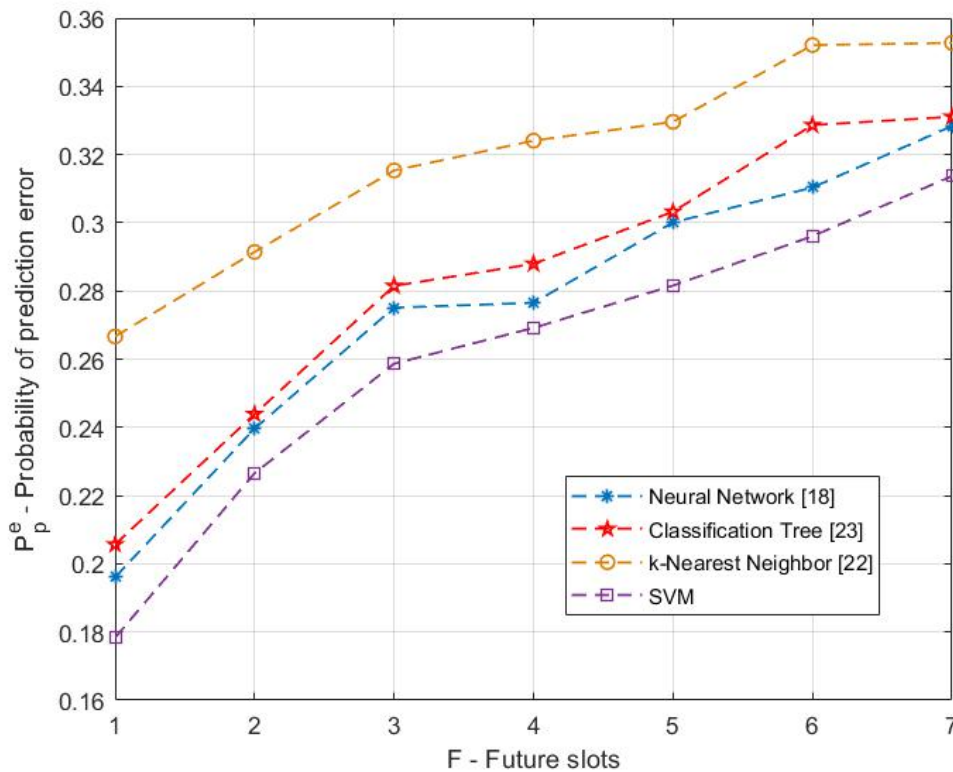


Fig.4. The performance of predictors measured in terms of the probability of prediction error as a function of the number of predicted future slots. The traffic density for all simulations was set to $\rho=0.65$.

A point that weighed heavily in favor of choosing the SVM tool is that it always converges to the optimal result [26]. This is not guaranteed by the Neural Network, the second best ML algorithm in terms of probability of wrong detection. Another point that was taken into consideration in our analysis was the computational cost. In our study, both the neural network algorithm and the SVM algorithm spent a very similar amount of time for its execution in the training phase, with a discrete difference in favor of the SVM ML algorithm. In the data validation phase (execution of the real data) the SVM technique operated about 17% faster than the neural network algorithm. Finally, to the best of our knowledge the authors are unaware of any multi-step-ahead prediction application with SVM.

Predictor performance is measured in terms of the probability of prediction error P_e^P , as described in Section III-A. In [13], the authors use two scenarios with constant P_e^P for different traffic density values ρ . In the present work, however, it was shown that the probability of prediction error P_e^P

depends on traffic density. This is illustrated in Figure 5, where P_e^p is non-linear and the least favorable cases correspond to Rayleigh fading scenarios. Increasing traffic density alone leads to a variation of approximately 160% in the probability of prediction error. The scenario used in the present study is therefore more realistic as it takes into account the dependence of predictor performance on traffic.

Figure 5 shows the probability of prediction error (which is the performance benchmark of the predictor) as a function of traffic density. For the cases where the traffic density is very high or very low, the predictor has an excellent performance obtaining results very similar, although different, for the two curves. In both cases, the traffic density is extremely high or extremely low and the uncertainty of future states is low. The predictor tends to be more accurate in these cases because these scenarios are more static. The fact that the probability of error of the SVM predictor is approximately 10% in both cases is associated with the residual error equation (2) being defined as $R_{emp}(f) = 0.1$. The predictor achieves this same level of performance, although the Rayleigh channel causes more dynamics in the traffic pattern, showing excellent performance of the predictor. In cases where the traffic density is close to $\rho = 0.5$, the uncertainty of the predictor assumes the highest level since there the scenarios are more dynamical.

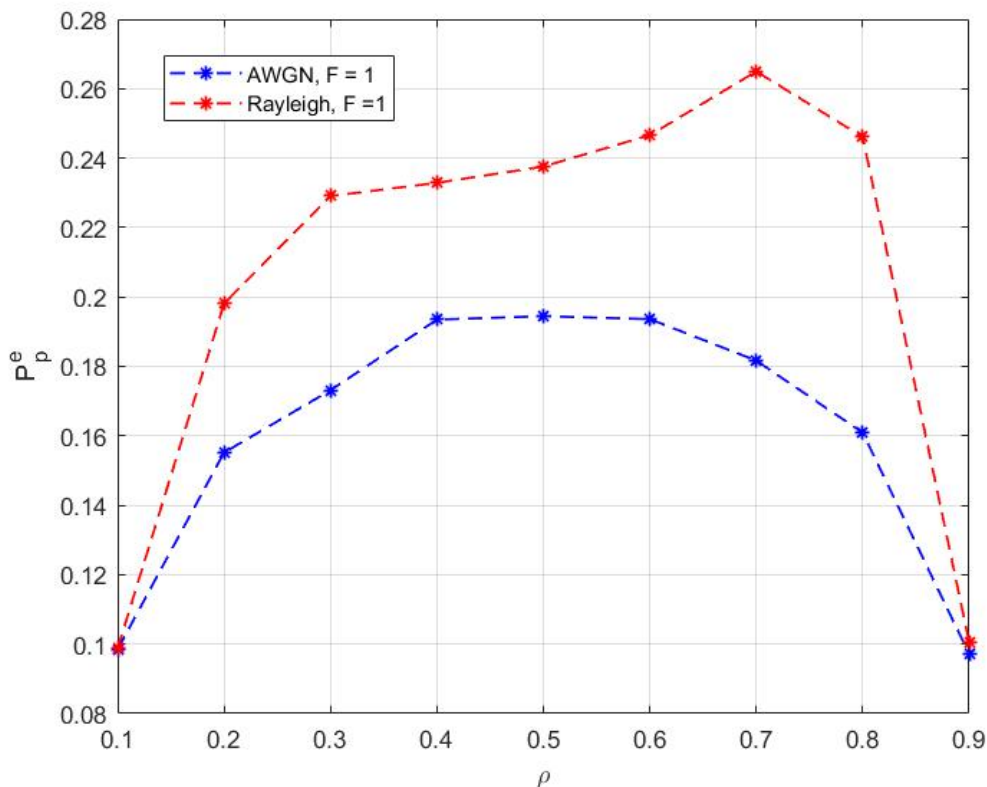


Fig. 5. Imperfect prediction as a function of traffic density, with \mathbf{N} constant.

B. The appropriate number of samples for the Rayleigh fading channel

Figure 6 shows the very high degree of similarity between the curves of P_d as a function of N generated by the equations in [34] and equations (9) and (13) in the present paper. There is a minimal difference between the calculated number of samples for both cases. In computational terms it was observed that computing N using (13) is less complex and 4.54 times faster than using equation (4.2) of [34].

Performance gains can be achieved during the sensing stage when the correct number of samples is used in the system design. In the case of the Rayleigh channel, there was an increase in energy-detector performance when we used the specific design for this channel type.

Figure 7 shows two simulations for detection probability values as a function of SNR. In the first simulation we used the design for an AWGN channel in a Rayleigh fading channel, while in the second simulation the design specifically intended for a Rayleigh fading scenario was used.

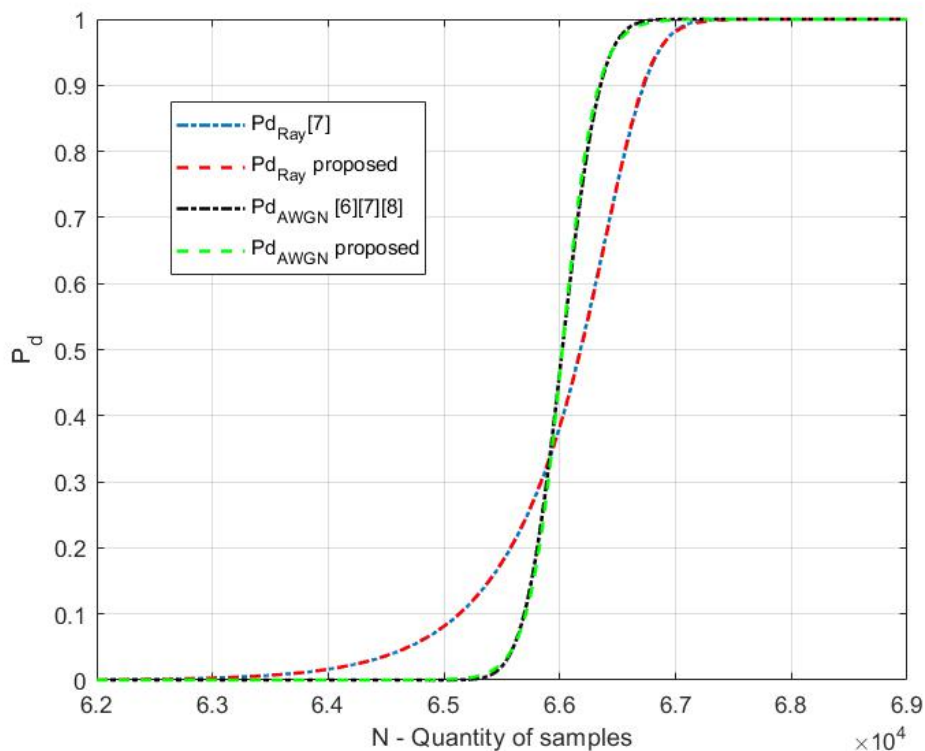


Fig. 6. Theoretical detection probability as a function of number of samples for an AWGN channel and a Rayleigh fading channel for the proposed approach and the literature.

For design values of $P_f = 0.1$, $P_d = 0.9$, and $\gamma = -20\text{ dB}$ in an AWGN channel, $N = 66,351$ samples were used. When this number of samples was used in a fading scenario, the performance of the energy detector was adversely affected. In this unfavorable case, the detection probability was approximately $P_d = 0.8$, corresponding to a performance loss of approximately 12%. This result shows the importance of analyzing fading scenarios. If only the project for the AWGN was considered but the channel is Rayleigh, the results would show a degradation in the performance of

the energy detector. The simulated detection probability $P_d = 0.9$ when the correct design for a Rayleigh channel was used can also be seen in Figure 7. The correct number of samples $N_{Ray} = 66,718$ for a Rayleigh channel was determined from (13).

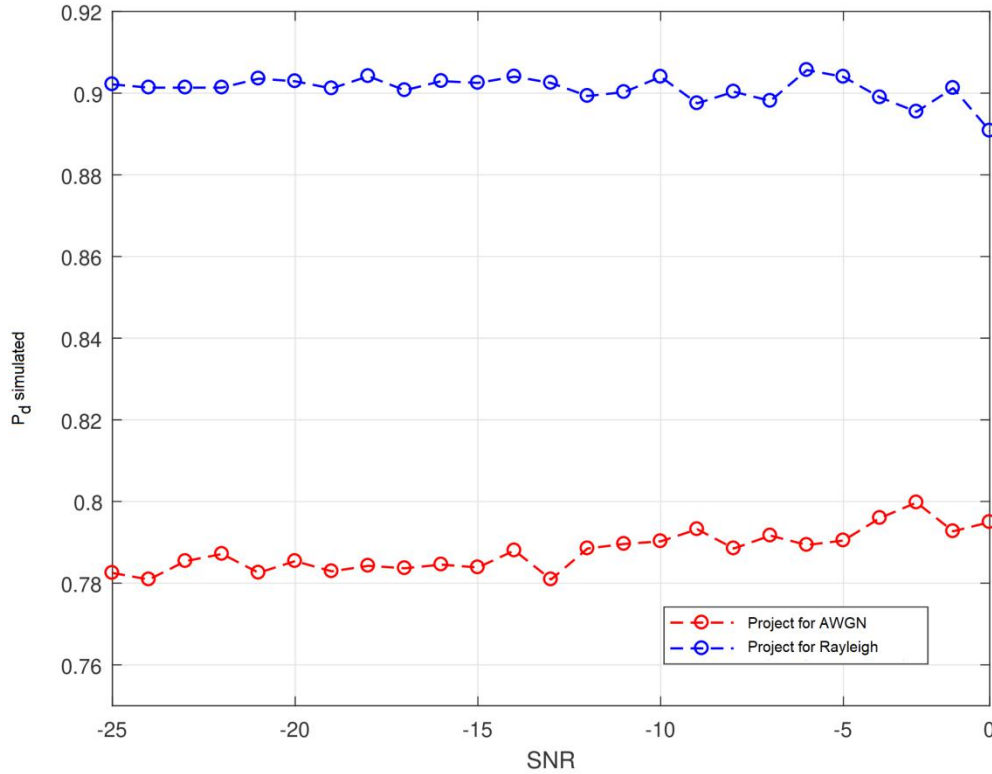


Fig. 7. Simulated detection probability as a function of SNR in a Rayleigh fading channel.

C. Channel Scheduling

In this simulation we used the specific design for a Rayleigh channel with a past observation window of $L = 10$, future observation window of $F = 5$ and traffic density values ranging from $\rho = 0.1$ to $\rho = 0.9$. Figure 8 shows a comparison of the amount of channel handoff with and without multi-step-ahead scheduling, which was described in Section III-E. In the scheduling scheme, the predictor uses the multi-step-ahead algorithm to choose the channel with the largest number of future idle slots. For low values of ρ most channels are idle, and when no decision criterion is used by the SU there is a large amount of channel handoff. When the multi-step-ahead prediction scheme is used, channel handoff is limited for low values of ρ and increases as ρ increases, but decreases when $\rho = 0.9$. For ease of interpretation, the amount of channel handoff was normalized by dividing the number of channel hops by the number of slots analyzed. For $\rho = 0.6$ there was a saving of up to 42% in the number of channel hops when multi-step-ahead scheduling was used, and for $\rho = 0.5$ the result of Figure 8 was 51%. When the traffic density is close to $\rho = 0.9$, it is possible to observe that the number of handoff is very low and the two curves are very close. In this case, most channels

are occupied during several time intervals. The SU does not perform the transmission because it is most often in the stop state. The two algorithms perform similarly in this extreme case and the network throughput is severely affected by the high traffic density

D. Throughput of the Cognitive Radio Network

Figure 9 shows the normalized throughput R_{Norm} as a function of traffic density for an AWGN scenario. Two simulations were performed for $F = 1$: one without channel scheduling and one with scheduling, as proposed in Section III-E. Two simulations were also performed for $F = 3$. In all four simulations the same number of past points ($L = 15$) was used. Curves of the theoretical normalized capacity for an AWGN channel with $P_e^p = 0.2$ obtained using the approach described in Section III-D are also shown. All the simulations yielded curves very close to the theoretical curves, and when channel scheduling was used there was an increase in network throughput. When channel with the most probability to be idle is chosen, there are fewer interruptions due to channel switching. In the case of $F = 1$ the scheduling algorithm gives preference to the channel that was used in the past state, avoiding interruptions.

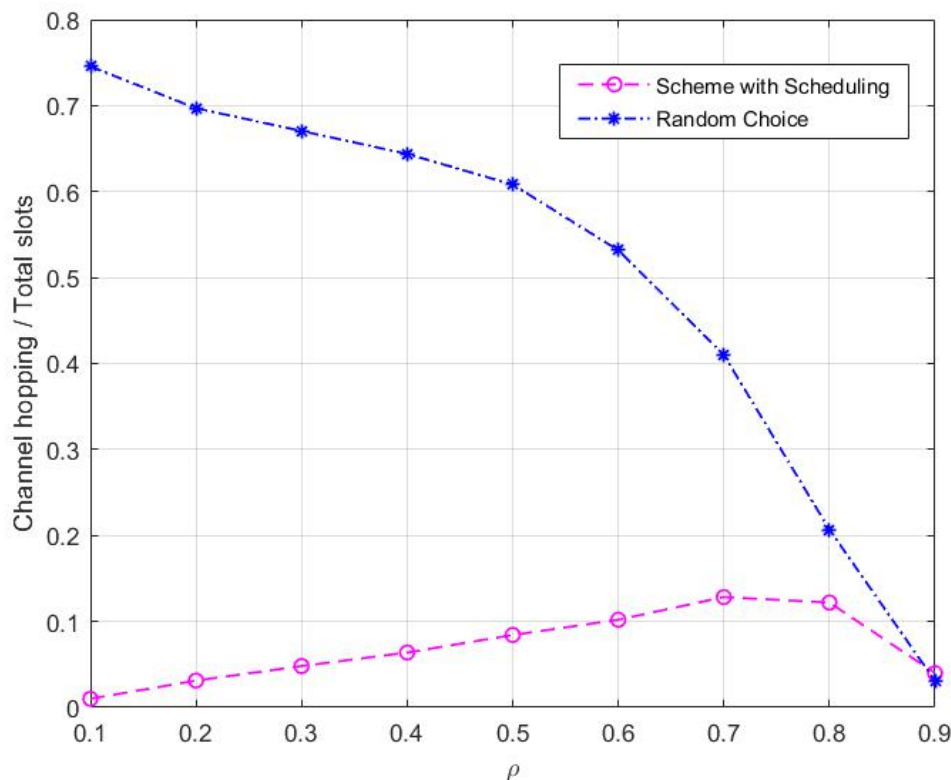


Fig. 8. Comparison of the amount of channel handoff with and without multi-step-ahead scheduling.

A gain in terms of throughput is observed when multi-step-ahead prediction is used. Figure 9 shows that the curve for $F = 3$ with scheduling was extremely close to the curve for $F = 1$ without

scheduling. The curve for $F = 1$ is also generated in [13] which analyses the case for single step prediction. For $\rho = 0.5$ the increase in throughput was about 4%. This increase is a result of multi-step-ahead prediction based on the structure proposed in Section III, which ensures an increase in the effective transmission time. Figure 9 also compare the cases with scheduling and without scheduling. In the case of multi-step prediction without scheduling the average probability of prediction error is slightly greater than the case with single step prediction due the error accumulation problem. The scheduling algorithm implemented in the SU improves the cognitive network throughput performance; the factor that contributes to this increase is that the channel-scheduling algorithm always chooses the channel with the greatest number of predicted idle slots. Thus, the transmission has the least amount of hops possible.

In the case of multi-step-ahead prediction, accumulation of error causes the SU predictor to perform slightly less than the single-step-ahead predictor for $\rho < 0.3$. For multi-step, the algorithm tries unsuccessfully to generate scheduling, ultimately generating unnecessary channel hopping. These changes cause loss of performance in the network throughput. The one-step forward algorithm has a more robust prediction and less channel change for these cases. For the other cases from $\rho \geq 0.3$ there is a performance gain by the multi-step-ahead prediction. As much as there is a problem of error accumulation, the algorithm can choose more coherently channels with the largest number of free channels, through scheduling.

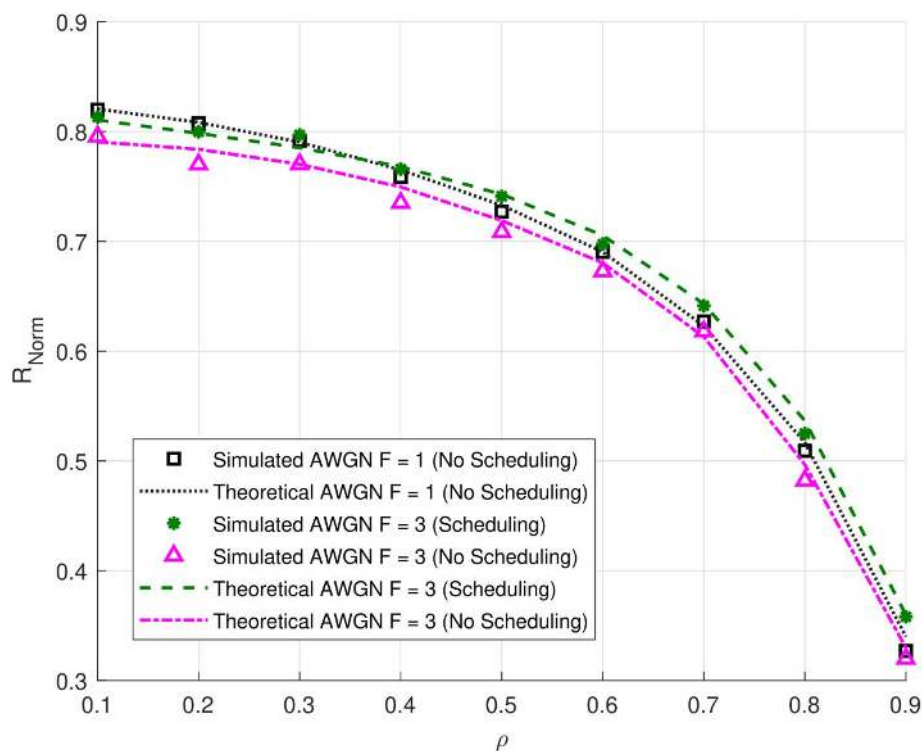


Fig. 9. Normalized throughput R_{Norm} of the CR network for the AWGN scenario with $F = 1$ and $F = 3$.

Figure 10 shows normalized throughput as a function of traffic density in a Rayleigh fading scenario. Two simulations were performed for $F = 1$ (with and without scheduling) and two for $F = 3$ (with and without scheduling).

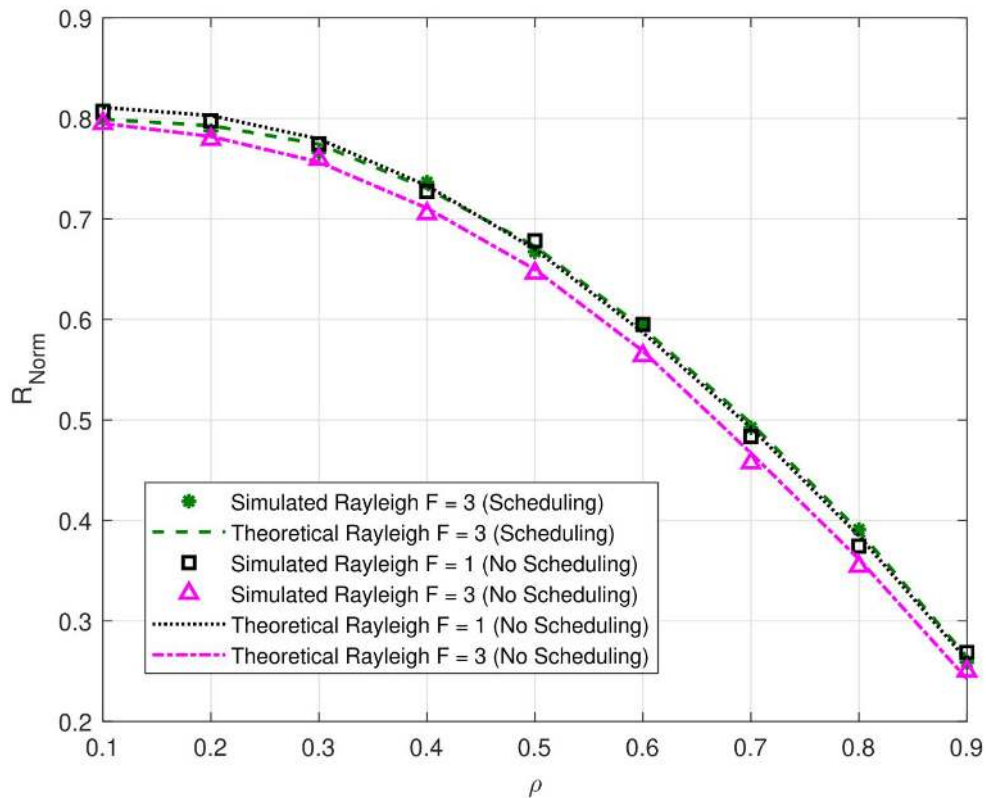


Fig. 10. Normalized throughput R_{Norm} of the CR network for Rayleigh fading scenarios with $F = 1$ and $F = 3$.

The theoretical curves for the Rayleigh channel based on these properties are also shown. As in the previous simulations, a value of $P_e^P = 0.2$ was used. Also, as in Figure 9, the simulated curves are very close to the theoretical curves. Again, channel scheduling produced gains in terms of throughput. There is an increase of approximately 3% for $\rho = 0.5$ for both $F = 1$ and $F = 3$. The result with scheduling for $F = 3$ is very close to the result without scheduling for $F = 1$. The gains in Figure 10 are significantly lower than those in Figure 9 because the Rayleigh channel is subject to greater signal degradation than the AWGN channel, and the performance of the predictor was adversely affected. The results for Rayleigh channel with single slot prediction and multi-step prediction were not explored in literature, the comparison is only realized with the AWGN case.

The results presented in this Section showed significant improvements when compared to some results presented in the literature. First, an improvement of up to 4% in the normalized transfer rate of the cognitive radio network (secondary user network) was found when using the prediction with multi-step ahead compared to the work [13]. This result was possible because the enhanced structure of the SU framework is different from what had already been published in the literature. Our work

also presented a channel scheduling scheme for multiple steps that had not been explored in the literature, this approach allowed the SU to choose the channel that has the best availability for the longest time. We also used the Support Vector Machine (SVM) machine learning technique, which had a lower probability of prediction error compared to other techniques found in the literature [18], [22] and [23]. This ML algorithm achieved the best results not only for one point in the future but for multiple points in the future. Finally, we present a new way of finding the probability of detection of the energy detector for the channel with Rayleigh fading that is simpler and faster to perform than the equation found in [34].

V. CONCLUSIONS

This work focused on the formulation of multi-step-ahead spectrum-prediction schemes for fading channels. Innovative analytical expressions for the probability of a cognitive radio network with multiple channels and multi-step-ahead prediction are theoretically derived. The energy-detector sensing technique was analyzed for the Rayleigh fading scenario, and a new detection-probability equation for low-SNR scenarios was derived. Using this equation, the exact number of samples required to meet the system design requirements was calculated. This approach also resulted in improvements in the throughput of the CR network. Using multi-step-ahead an algorithm was explored which permit SU prioritize channels that are most likely to be idle longer so there is a saving in terms of hand-off periods for users. The proposed approach also reduces the number of repetitions of time spent on predictions in SU frame, generating throughput gains for the entire network. Excellent levels of prediction error were achieved using the SVM technique. Prediction error was found to depend on PU traffic, which was included as a variable parameter to improve the simulation. The main suggestions for future research include realize an energy-efficient multi-step prediction algorithm for different fading scenarios and propose a system model were SUs employing multi-step prediction can cooperate with each other providing enhanced network throughput. The continuous study of efficient techniques for the use of the spectrum opens space for new practical models of cognitive radio to be implemented in new generations of wireless communication systems.

REFERENCES

- [1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Pers. Comm.*, vol. 6, no. 4, pp. 13–18, Aug., 1999, DOI. 10.1109.98.788210.
- [2] IEEE Draft Standard for Information Technology -"Telecommunications and information exchange between systems - Wireless Regional Area Networks (WRAN) - Specific requirements - Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV Bands," in IEEE P802.22/D1.0, December 2010 , vol., no., pp.1-598, 20 Dec. 2010
- [3] Z. Wei and B. Hu, "A Fair Multi-Channel Assignment Algorithm With Practical Implementation in Distributed Cognitive Radio Networks," *IEEE Access*, vol. 6, pp. 14255-14267, Mar., 2018, DOI. 10.1109.ACCESS.2018.2808479.
- [4] A. Goldsmith, S. A. Jafar, I. Maric and S. Srinivasa, "Breaking Spectrum Gridlock With Cognitive Radios: An Information Theoretic Perspective," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, May, 2009, DOI. 10.1109.JPROC.2009.2015717.
- [5] M. A. Abdulsattar and Z. A. Hussein, "Energy detection technique for spectrum sensing in cognitive radio: a survey," *International Journal of Computer Networks and Communications*, vol. 4, no. 5, pp. 223-242, Sep., 2012, DOI. 10.5121.ijcnc.2012.4514.
- [6] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523-531, Sep., 1967, DOI.10.1109.PROC.1967.5573.

- [7] S. Atapattu, C. Tellambura and H. Jiang, "Conventional Energy Detector," in Energy detection for spectrum sensing in cognitive radio, 1st ed. New York, NY, US: Springer, 2014, ch. 2, sec. 3, pp. 11–24.
- [8] Y. Liang, Y. Zeng, E. C. Y. Peh and A. T. Hoang, "Sensing-Throughput Tradeoff for Cognitive Radio Networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1326–1337, Apr., 2008, DOI. 10.1109.TWC.2008.060869.
- [9] S. Atapattu, C. Tellambura and H. Jiang, "Spectrum Sensing via Energy Detector in Low SNR," *2011 IEEE International Conference on Communications (ICC)*, Kyoto, pp. 1–5, DOI. 10.1109.icc.2011.5963316.
- [10] S. Nandakumar et al., "Efficient Spectrum Management Techniques for Cognitive Radio Networks for Proximity Service," *IEEE Access*, vol. 7, pp. 43795–43805, Apr., 2019, DOI. 10.1109.ACCESS.2019.2906469.
- [11] A. M. Masri, C. Chiasserini, C. Casetti and A. Perotti, "Common control channel allocation in cognitive radio networks through UWB communication," *Journal of Communications and Networks*, vol. 14, no. 6, pp. 710–718, Dec., 2012, DOI. 10.1109.JCN.2012.00037.
- [12] A. Shahid et al., "CSIT: channel state and idle time predictor using a neural network for cognitive LTE-Advanced network," *Eurasip Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 203–219, Dec., 2013, DOI. 10.1186.1687-1499-2013-203.
- [13] J. Yang and H. Zhao, "Enhanced Throughput of Cognitive Radio Networks by Imperfect Spectrum Prediction," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1738–1741, Oct., 2015, DOI. 10.1109.LCOMM.2015.2442571.
- [14] G. Ding et al., "On the limits of predictability in real-world radio spectrum state dynamics: from entropy theory to 5G spectrum sharing," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 178–183, Jul., 2015, DOI. 10.1109.MCOM.2015.7158283
- [15] X. Xing, T. Jing, W. Cheng, Y. Huo and X. Cheng, "Spectrum prediction in cognitive radio networks," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 90–96, Apr., 2013, DOI. 10.1109.MWC.2013.6507399
- [16] G. Ding et al., "Spectrum Inference in Cognitive Radio Networks: Algorithms and Applications," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 1, pp. 150–182, Firstquarter, 2018, DOI.10.1109.COMST.2017.2751058
- [17] X. Chen and J. Yang, "A spectrum prediction-based frequency band pre-selection over deteriorating HF electromagnetic environment," *China Communications*, vol. 15, no. 9, pp. 10–24, Sep., 2018, DOI. 10.1109.CC.2018.8456448
- [18] V. K. Tumuluru, P. Wang and D. Niyato, "A Neural Network Based Spectrum Prediction Scheme for Cognitive Radio," *2010 IEEE International Conference on Communications*, pp. 1–5, Cape Town, May, 2010, DOI. 10.1109.ICC.2010.5502348.
- [19] Z. Tang and S. Li, "Deep Recurrent Neural Network for Multiple Time slot Frequency Spectrum Predictions of Cognitive Radio," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 6, pp. 3029–3045, Oct., 2017, DOI. 10.3837.tiis.2017.06.013.
- [20] N. Shamsi, A. Mousavinia and H. Amirpour, "A channel state prediction for multi-secondary users in a cognitive radio based on neural network," *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, pp. 200–203, Ankara, Nov., 2013, DOI. 10.1109.ICECCO.2013.6718263.
- [21] A. Agarwal, S. Dubey, M. A. Khan, R. Gangopadhyay and S. Debnath, "Learning based primary user activity prediction in cognitive radio networks for efficient dynamic spectrum access," *2016 International Conference on Signal Processing and Communications (SPCOM)*, pp. 200–203, Bangalore, Jun., 2016, DOI. 10.1109.SPCOM.2016.7746632.
- [22] Z. Chen, N. Guo, Z. Hu and R. C. Qiu, "Channel state prediction in cognitive radio, Part II: Single-user prediction," *2011 Proceedings of IEEE Southeastcon*, pp. 50–54, Nashville, TN, Mar. 2011, DOI. 10.1109.SECON.2011.5752904.
- [23] Soltani, S., and Mutka, M. W., "A decision tree cognitive routing scheme for cognitive radio mesh networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 10, pp. 1405–1417. 2015, DOI. 10.1002/wcm.2418
- [24] Canavitsas, A., Silva Mello, and M. Grivet, "Spectral Vacancies Prediction Method for Cognitive Radio Applications," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 15, no. 1, pp. 18–29, Mar., 2016, DOI. 10.1590.2179.10742016v15i1581
- [25] X. Chen, J. Yang and G. Ding, "Minimum Bayesian Risk Based Robust Spectrum Prediction in the Presence of Sensing Errors," *IEEE Access*, vol. 6, pp. 29611–29625, May, 2018, DOI. 10.1109/ACCESS.2018.2836940
- [26] N. Sapankevych and R. Sankar, "Time Series Prediction Using Support Vector Machines: A Survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, May, 2009, DOI. 10.1109/MCI.2009.932254
- [27] R. Min et al. "Interference avoidance based on multi-step-ahead prediction for cognitive radio," *2008 11th IEEE Singapore International Conference on Communication Systems*, Guangzhou, Nov., 2008, DOI. 10.1109.ICCS.2008.4737177.
- [28] B. Shawel et al. "Convolutional LSTM-based Long-Term Spectrum Prediction for Dynamic Spectrum Access," *2019 27th European Signal Processing Conference (EUSIPCO)*, A Coruna, Sep., 2019, DOI. 10.23919.EUSIPCO.2019.8902956.
- [29] J. Sun et al. "Long-Term Spectrum State Prediction: An Image Inference Perspective," *IEEE Access*, vol. 6, pp. 43489–43498, Jul., 2018, DOI. 10.1109/ACCESS.2018.2861798
- [30] C. Ge, Z. Wang and X. Zhang, "Robust Long-Term Spectrum Prediction With Missing Values and Sparse Anomalies," *IEEE Access*, vol. 7, pp.16655–16664, Jan., 2019, DOI. 10.1109/ACCESS.2018.2889161
- [31] S. Bayhan and F. Alagoz, "Scheduling in Centralized Cognitive Radio Networks for Energy Efficiency," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 582–595, Feb., 2013, DOI. 10.1109.TVT.2012.2225650
- [32] P. Chang et al., "Performance analysis of channel switching with various bandwidths in cognitive radio," *2013 The twelfth International Conference on Networks (ICN)*, pp. 29–33, Seville, Jan., 2013, DOI. 10.1109.ICECCO.2013.6718263.

- [33] V. N. Vapnik, "Methods of Pattern Recognition," in *The Nature of Statistical Learning Theory*, 1th ed. New York, NY, US: Springer-Verlag, 1995, ch. 5 , sec. 1 , pp. 123–167.
- [34] S. Atapattu, C. Tellambura and H. Jiang, "Performance Measurements," in *Energy detection for spectrum sensing in cognitive radio*, 1st ed. New York, NY, US: Springer, 2014, ch. 4 , sec. 1 , pp. 41–59.
- [35] Allwein, E., Schapire, R. and Singer Y. "Reducing multiclass to binary: A unifying approach for margin classifiers." *Journal of Machine Learning Research*, vol. 1, pp. 113–141, Dec., 2000.
- [36] M. Abramowitz and I. A. Stegun, "Repeated Integrals of the Error Function," in *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, 10th printing. San Washington, D.C, US: National Bureau of Standards Applied Mathematics Series - 55, 1972, ch. 7 , sec. 1.25 , pp. 299.
- [37] I. S. Gradshteyn and I. M. Ryzhik, "Hypergeometric Functions," in *Tables of Integrals, Series, and Products*, 7th ed. San Diego, CA, US: Elsevier Academic Press, 2007, ch. 9 , sec. 1 , pp. 1005–1048.
- [38] K. D. Rao, "Introduction," in *Channel coding techniques for wireless communications*, 1st ed. New Delhi, India, Springer, 2015, ch. 1 , sec. 4 , pp. 1–20.
- [39] V. Kuhn, *Wireless communications over MIMO channels: applications to CDMA and multiple antenna systems*, 1st ed. Chichester, UK, John Wiley et Sons, 2006, ch. 2 , sec. 2 , pp. 51–90.
- [40] Mushtaq, M. T., Khan, I., Khan, M. S., and Koudelka, O. "Signal detection for QPSK based cognitive radio systems using support vector machines." *Radioengineering*, vol. 24, no. 1, pp 192-198. Apr. 2015. DOI: DOI. 10.13164/re.2015.0192