UNIVERSIDADE FEDERAL DE LAVRAS

OTÁVIO NEVES LARA

METHODOLOGICAL COMPARISON OF MACHINE LEARNING TECHNIQUES TO IMPROVE ADAPTABILITY AND REDUCE THE HANDOFF RATE IN COGNITIVE RADIOS

LAVRAS – MG

2021

OTÁVIO NEVES LARA

METHODOLOGICAL COMPARISON OF MACHINE LEARNING TECHNIQUES TO IMPROVE ADAPTABILITY AND REDUCE THE HANDOFF RATE IN COGNITIVE RADIOS

Dissertation presented to the Universidade Federal de Lavras, as part of the requirements of the Postgraduate Program in Computer Science, concentration area in Computer Networks, in order to obtain the title of Master.

Prof. DSc. Luiz Henrique Andrade Correia
Orientador

Prof. DSc. Erick Galani Maziero
Coorientador

LAVRAS – MG
2021

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo próprio autor

Lara, Otávio
Methodological Comparison of Machine Learning Techniques to Improve Adaptability and Reduce the Handoff Rate in Cognitive Radios / Otávio Neves Lara. – Lavras : UFLA, 2021.
54 p. : il.

Dissertação(Mestrado)–Universidade Federal de Lavras, 2021.
Orientador: Prof. DSc. Luiz Henrique Andrade Correia.
Coorientador: Prof. DSc. Erick Galani Maziero.
Bibliografia.

1. Computer Networks. 2. Artificial Intelligence. I. Correia, Luiz Henrique Andrade. II. Maziero, Erick Galani.

CDD-808.066

OTÁVIO NEVES LARA

METHODOLOGICAL COMPARISON OF MACHINE LEARNING TECHNIQUES TO IMPROVE ADAPTABILITY AND REDUCE THE HANDOFF RATE IN COGNITIVE RADIOS

Dissertation presented to the Universidade Federal de Lavras, as part of the requirements of the Postgraduate Program in Computer Science, concentration area in Computer Networks, in order to obtain the title of Master.

APROVADA em 30 de Abril de 2021.

Prof. DSc. Daniel Fernandes Macedo	UFMG
Prof. DSc. Demostenes Zegarra Rodriguez	UFLA

Prof. DSc. Luiz Henrique Andrade Correia
Orientador

Prof. DSc. Erick Galani Maziero
Co-Orientador

LAVRAS – MG
2021

Dedico esse trabalho à todos os professores que passaram e passarão pela minha vida.

AGRADECIMENTOS

Agradecimento especial ao suporte da minha família, aos meus orientadores e coorientadores durante a pós graduação, e à CAPES.

“Nesse mundo a gente guarda o que é de comer, mas não guarda o que é de fazer”

Ávila, Maura Júlia

ABSTRACT

A large number of devices connected in the Industrial, Scientific & Medical (ISM) bands, in large urban centers, makes interference between them inevitable. Analyzing this, the Federal Communications Commission (FCC) assembled in 2002 a team to improve the policy of separation of the electromagnetic spectrum: the Spectrum Police Task Force (SPTF). Then the Cognitive Radio (CR) were proposed as a solution, allowing secondary users to connect on frequencies reserved for primary users, with low interference between them. One of the functions of CR is the automatic selection of channels in the electromagnetic spectrum. Several algorithms have been proposed to predict which will be the next channel, but few are concerned with the geographic adaptability of the model and the number of handoffs that the radio does. In this work, we propose a methodological comparison between the CRF and Q-Learning, to analyze the adaptability of these two distinct geographic local techniques, maintaining precision and a low handoff rate. Testing on Wi-Fi frequencies the CRF shows more adaptive, due its temporal windows, on average with 97.6% adaptability versus 95.68% for Q-Learning. In addition, a CRF handoff rate remained below 0.1% in all locations and frequencies, against an average of 5.8% for Q-Learning.

Keywords: Cognitive Radio. Spectrum decision. Machine Learning.

RESUMO

Uma grande quantidade de dispositivos conectados nas bandas Industriais, Científicas e Médicas (ICMs), em grandes centros urbanos, faz com que a interferência entre eles seja inevitável. Considerando isso, a *Federal Communication Commicion (FCC)* montou em 2002 uma equipe para melhorar a política de gerência do espectro eletromagnético: a *Spectrum Policy Task Force (SPTF)*. Foi então que os Rádios Cognitivos (RCs) foram propostos como uma solução, permitindo que usuários secundários se conectem em frequências reservadas para usuários primários, com baixa interferência entre eles. Uma das funções dos RCs é a seleção automática de canais do espectro eletromagnético. Muitos algoritmos foram propostos para prever qual será o próximo canal, porém poucos se preocupam com a adaptabilidade geográfica do modelo e quantidade de *handoffs* que o rádio faz. Neste trabalho, propomos uma comparação metodológica entre os algoritmos *Conditional Random Fields (CRF)* e *Q-Learning*, para analisar a capacidade de adaptação dessas duas técnicas em locais geográficos distintos, mantendo a precisão e uma baixa taxa de *handoff*. Testando nas frequências do Wi-Fi o CRF se mostrou em média mais adaptativo, por considerar uma janela temporal, com 97,6% de adaptabilidade contra 95,68% do *Q-Learning*. Ademais, a taxa de *handoff* do CRF permaneceu inferior a 0,1% em todos locais e frequências, contra uma média de 5,8% do *Q-Learning*.

Palavras-chave: Rádios Cognitivos. Decisão do Espectro. Aprendizado de Máquina.

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CI	Confidence Interval
CPU	Central Processing Unit
CR	Cognitive Radio
CRF	Conditional Random Fields
CRN	Cognitive Radio Network
DRL	Deep Reinforcement Learning
DSA	Dynamic Spectrum Access
DSM	Dynamic Spectrum Management
DSP	Digital Signal Processors
FCC	Federal Communications Commission
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FUTEBOL	Federated Union of Telecommunications Research Facilities for a EU-Brazil Open Laboratory
GRC	GNU Radio Companion
GRUBI-COM	Grupo de Redes Ubíquas e Comunicação
HMM	Hidden Markov Model
IBGE	Instituto Brasileiro de Geografia e Estatística
IEEE	Institute of Electrical and Electronic Engineers

IRIS . . . . Implementing Radio in Software

ISM . . . . Industrial, Scientific & Medical

LabVIEW . . . . Laboratory Virtual Instrument Engineering Workbench

MAC . . . . Medium Access Control

MDP . . . . Markov Decision Process

ML . . . . Machine Learning

MLP . . . . Multilayer Perceptron

NSF . . . . National Science Foundation

OOP . . . . Oriented object programming

OOT . . . . Out Of Tree

OS . . . . Operating System

OSA . . . . Opportunistic Spectrum Access

OSSIE . . . . Open-Source SCA Implementation - Embedded

PUs . . . . Primary Users

QoE . . . . Quality of Experience

QoS . . . . Quality of Service

RF . . . . Radio Frequency

RL . . . . Reinforcement Learning

RnF . . . . Random Forest

SD . . . . Standard Deviation

SDR . . . . Software Defined Radio

SHF . . . . Super High Frequency

SHs . . . . Spectrum Holes

SPTF . . . . Spectrum Police Task Force

SUs . . . . Secondary Users

UCB . . . . Upper Confidence Bound

UFLA . . . . Universidade Federal de Lavras

UHD . . . . USRP Hardware Driver

USRP . . . . Universal Software Radio Peripheral

VIs . . . . Virtual Instruments

XML . . . . Extensible Markup Language

CONTENTS

1 INTRODUCTION

The Industrial, Scientific & Medical (ISM)[1] frequency bands are becoming busy. Some bands like Wi-Fi (2.4 GHz and 5 GHz) are almost 90% busy in some areas (SHAIKH et al., 2016). Moreover, the spectrum has been wasted all these years, there are about 30 GHz of usable frequency, called Super High Frequency (SHF), spectrum which could be used for data transmission and just a few percentages are realy used. In 2002 the Federal Communications Commission (FCC) allow Secondary Users (SUs) with unlicensed devices to connect in Primary Users (PUs) frequencies, without the interference of SUs in PUs, also known as Opportunistic Spectrum Access (OSA)(FCC, 2002). One solution is using Cognitive Radio (CR), which uses Artificial Intelligence (AI) to predict which frequency range will be free in the next time slot.

Implementing algorithms in hardware is a challenge, since circuits are not easily rewritable. In order to test techniques proposed for CR, it is simpler and faster when using Software Defined Radio (SDR). SDR are radios which can reconfigure their parameters through software (RONDEAU et al., 2004). The first architecture were defined by Mitola (1995), called Software Radios. SDR allows researchers fast concept testing, which rise the idea of transforming the legacy radios into smart radios, that are capable to be aware of the RF environment.

CR uses algorithms in the Spectrum Decision phase, allowing the choice between frequency bands without interfering between the radio devices. The main goal of CR is the awareness of the RF environment which implies adaptability in any context (MITOLA; MAGUIRE, 1999). The CR operation is divided into phases (CHOWDHURY; AKYILDIZ, 2008): *i)* spectrum sensing, *ii)* spectrum decision, *iii)* spectrum sharing, and *iv)* spectrum mobility. Spectrum sensing *i)* is the start of the working flow, with the radio collecting the variables of its environment such as power, noise floor, latency, and throughput. The spectrum decision phase *ii)* analyzes the data collected from the spectrum and choices to improve the future of the radio and its environment. After deciding the better parameter to reconfigure they must share *iii)* with all the Cognitive Radio Network (CRN) the chosen value. When all radios have the new frequency value, for example, all radios move *iv)* to the new chosen frequency.

The CR known to deal with the spectrum of frequencies is called Dynamic Spectrum Access (DSA). The goal is to allow Secondary Users (SUs) connect in the Primary Users (PUs)' frequencies without properly interfere in the frequency owners. In this work, the focus is to present solutions to test better models of spectrum decision. In the literature of CR there was not a standardization of the way that specturm decision models are tested.

[1] Anatel Resolution Nº 716

In some works like SIRCo framework proposed by Marques et al. (2016), were implemented a Artificial Neural Network (ANN) without testing in different locations. However, the spectrum is non-stationary environment (HUANG et al., 1998) and might have different behavior depending on the location where the radio is placed. Without adaptation, the algorithm might not converge when either the environment or time changes, this problem was noticed by the authors in the conclusion of Marques et al. (2016) work.

Beside, there is not a standardized test methodology to compare machine learning techniques for spectrum prediction, other literature works of CR do not use the same tests platform leading to a difficult results comparison. At Xing et al. (2013) the researchers test the Bayesian technique in a non-real simulated environment, considering the chance of SUs and PUs to use or leave a frequency based in a random exponential distribution. This simulation approach was also used in several works (WELLENS; RIIHIJARVI; MAHONEN, 2010) (MIN et al., 2011). Other approaches created their own framework using Software Defined Radio (SDR) (GONÇALVES, 2019). This allows tests in real environment and collect metrics such QoS, QoE, latency, jitters, throughput, and more. Also, In (MARQUES et al., 2016) and (PINTO; CORREIA, 2018) the authors tested Artificial Neural Networks, Hidden Markov Model, and Random Forest in a real Cognitive Radios Network (CRN) using the frequencies from 800 MHz to 5.8 GHz and tested with SDR in a real environment. However, it is important to train and test in more than one environment to evaluate the model's adaptability, because, that is the required scenario for commercial CRs.

This dissertation proposes a testing methodology to verify CR's adaptability in different locations furthermore AI algorithms that perform in the testing methodology. Were collected datasets of the Wi-Fi frequencies (2.4 GHz and 5.8 GHz) at two different locations, the cities Varginha and Pouso Alegre, to be used in the testing methodology. The testing methodology general idea is to train the algorithms in one place and test in others more to collect the metrics of accuracy, handoff rate, and adaptability. The accuracy is the relationship between the mistakes and successes of free channel choices, the handoff rate is how much the cognitive radio chooses to change the channel, and the adaptability is the capability of the model to keep these two metrics when the location changes. A channel is considered free when the power detected in this channel is over the threshold, which is usually considered between -85 dBm to -75 dBm.

The goal is to test how much the algorithm can keep the value of the metrics without retraining. This methodology also does not require that the researcher travels to different locations to test this model, the idea is to create a dataset of radio frequencies collected in several places around the world.

1.1 Motivation

Primary Users (PUs) have their bandwidth ensured by the local regulatory communication organization of each country. Cognitive radios come for the increase of the spectrum access for Secondary Users (SUs) without interfering in PUs' Quality of Service (QoS).

During the literature research of this dissertation was noticed that it is not trivial to compare works of decision phase of CR. Each work tests the proposed model using a different approach. Some works implement a simulated environment using statistics methods considering a distributed exponential data behavior of CRs. Other works implement radios in real environments using SDR. However, a radio trained and tested in only one local does not evince if the model will keep the performance in a new location with contrasting characteristics. Furthermore, in this dissertation was demonstrated that the characteristic where the radio is placed could interfere a lot in it performance.

In the literature, there are many AI algorithms which are used at the spectrum decision phase of CR. The authors Bkassiny, Li e Jayaweera (2013) notice that techniques which are not thought to adapt in new geographic locations could have low performance when the CR is tested in another location. As result, the lack of a standardized testing methodology makes it difficult to compare techniques.

The search for better spectrum decision algorithms aims to improve performance to minimize interference between SUs and PUs and increase the number of spectrum bands used per area, making better use of the electromagnetic frequencies spectrum.

1.2 Problem Definition

With the development of cognitive radios, some points of the tests performed in the literature have been discussed. The main factor that negatively influences the development of commercial cognitive radio devices is the ability of models applied to the spectrum decision phase to adapt to different geographic locations.

The mobility and adaptability is the main metric that few articles of the literature achieved. A Cognitive Radio with mobility means that the radio is capable to change its location, and adaptability is the capability of this radio to re-adapt in this new environment. In other words, keep the accuracy without interfering in the of the chosen frequency range.

The most important hypothesis developed in this dissertation questions how and how much the geographic mobility influences the quality and performance of models used in the decision phase of cognitive radio spectrum. In other words, it is relevant to analyze whether the format of training and testing of learning models used by the works presented in the literature presents a comparable and applicable result in any scenario, be it chaotic or relaxed.

Although there is a large, sparsely occupied portion of the spectrum today, it is necessary to think about how much a more competitive environment, such as the Wi-Fi frequency bands, can make more generic models capable of adapting to environments that are less or as chaotic. Several works were developed with tests in large spectrum bands such as 3 GHz (RONDEAU et al., 2004) or more than 5 GHz (PINTO; CORREIA, 2018), which can make the model comfortable in choosing bands with no use during training and improve usability.

1.3 Goals

The general goal of this project is the development of a standardized testing methodology and the use of predictive models for the decision phase of CR. The test methodology allows researchers to identify the level of geographic adaptability and compare it with other works, which use the same technique, the performance and the accuracy of the model prediction. To reach this were collected datasets of the electromagnetic spectrum in the Wi-Fi frequencies (2.4 GHz and 5.8 GHz) in two different cities.

Were chosen two predictive models to test if the methodology indicators point for the better model. These models were applied to the test methodology and collecting the metrics of accuracy, handoff rate, and adaptability. The accuracy shows how much the model chooses the best free channel. The handoff rate allows the analysis of how much the model requires the CR to change its frequency. The handoff rate is undesired, the models must minimize it to reduce the spectrum mobility, the most expensive phase of CRN. Adaptability is a second-level metric, that is, it is necessary to collect the metric of accuracy and handoff rate in both locations, the adaptability allows the comparison of how much the accuracy and handoff rate persist in different environments.

At the end of this process, discuss the results and analyze if the test methodology achieves the goal of observing the adaptability, comparing the results, and the analysis of the collected data.

1.4 Justification

The number of devices connected in IEEE802.11 bands have been growing and hence the interference between them (SHAIKH et al., 2016). Also, at the USA, the Federal Communications Commission (FCC) has been approved in 2002 the use of unlicensed devices in licensed bands (KOLODZY; AVOID-

ANCE, 2002). Since then, researchers are developing techniques to control the spectrum access dynamically (WU et al., 2010).

Many Machine Learning (ML) approaches were tested in CR and shown high accuracy. However, only a few techniques were concerned about adaptability of the model in different geographic environments or the number of handoffs made by the model's predictions. Moreover, it is a challenge to compare the results of CR works, because each one was tested in a different spectrum size, spectrum portion, and some techniques were not tested in real spectrum, only using generated datasets of power. Our approach proposes a testing methodology, using the Wi-Fi bands 2.4 GHz and 5.8 GHz, due its high usage in urban center, to test predictive models.

1.5 Contributions

Targeting the goals and motivations of this work, we propose a testing methodology and two predictive models, Conditional Random Fields (CRF) and Q-Learning, to reduce the number of handoffs and improve the geographic adaptability of the models in the CR decision phase.

The test methodology consists in evaluate models without the bias of the environment. Standardize how the predictive models are tested, allows a fair comparison of several models already tested (with the same methodology) in the literature. This is done by applying the models in a multi-environment, repetitive set of tests in a concurrent environment. In other words, train the model in a location and test in others reproducing each step several times to reach the degree of confidence of the model applied. The goal is to reach the geographic adaptability level of the model. The concurrent environment cited above are the Wi-Fi frequencies, 2.4 GHz and 5.8 GHz, which reach more than 90% of usage in some locations.

In order to evaluate the test methodology were used the Q-Learning and CRF as predictive models. The adaptability of both was compared and discussed at the end of this work.

1.6 Organization

This work is organized as follow. In Section 2, are present the concepts of Cognitive Radio and Machine Learning. In Section 3 are presented the related works. The Section 4 presents how the spectrum dataset was collected, the proposed methodology of tests, and the implemented algorithms. The Section 5 discusses the results of the algorithms in each frequency and local, after applied the tests methodology. In Section 6, we finish the dissertation presenting a brief conclusion about this and future works.

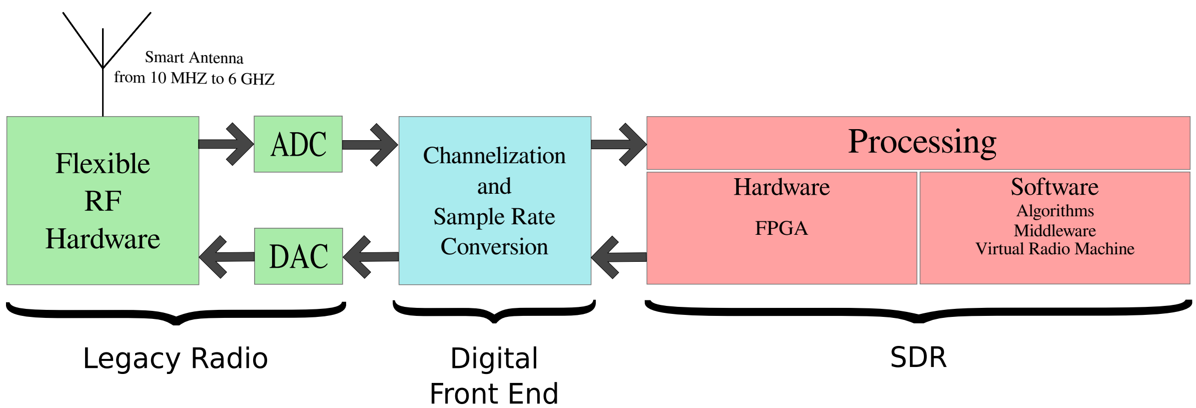
2 BACKGROUND

This section presents the definition of Software Defined Radio (SDR) and Cognitive Radio (CR); the explanation about the SDR tools and the GNU Radio; A explanation about some problems of SDR; The Machine Learning (ML) methods used in this work, Reinforcement Learning (RL) and Conditional Random Fields (CRF); At the end, a explanation about the hyperparameter optimizer used in this work, the IRace.

2.1 Software Defined Radios

Software Defined Radio (SDR) allows flexibility and changes the idea of hardware defined radios (once create, never changes) and let easy the implementation of algorithms to deal with radios parameters.

Figure 2.1 – SDR Schema



Font: author (2021)

The Figure 2.1 present the workflow of SDR and also where it is different from the legacy radio, how seen in the part signalized as legacy radios. After the signal demodulation the SDR has a front end which works like an interface between the radio and the software, changing parameters such as the sample rate to collect signals information. In the SDR part it is possible to use software to test the concept, and further an Field Programmable Gate Array (FPGA), speedup the software application.

There are many SDR tools to build the software, that will be present in the section 2.3. GNU Radio is a SDR tool that allow to use Python or C++ to develop an application. Python has known ML libraries that might be used within a SDR software. This open space to creation of a smarter radios, which could make decisions, or take actions based in the data collected from the Radio Frequency (RF) environment.

2.2 Cognitive Radios

How seen in Section 1.2, the FCC set the licensed frequencies free for SUs, it creates the motivation of utilization of the Cognitive Radio.

CR is a paradigm that attempts to opportunistically transmit in licensed frequencies, without affecting the PUs. The main goal of a CR is to predict the next busy spectrum aiming the not interference of SUs in PUs. CR uses SDR to creates intelligent radios that make decisions to avoid interference between devices and improve radio transmissions. Moreover, the problem of busy frequencies on ISM frequencies, such as 2.450 MHz, is that in some regions the usage of this band reached 90%. The Figure 2.2 of Shaikh et al. (2016) shows the usage of the 2.4 GHz in different metropolitan regions of the world, collecting surveys at Spain, Germany, Singapore, United Kingdom, and United States of America. More details may be seen in Shaikh et al. (2016).

In general, radios are connected in some specific frequencies as a communication pattern to make easier to connect with new devices, e.g. 5.8 GHz, in Wi-Fi IEEE802.11n devices. The growth of devices connected in ISM frequencies, raises together the interference in this bands of the spectrum (AKYILDIZ et al., 2008). Among the consequences of interference are package loss and traffic slowness, making the usage difficult. One solution proposed in the literature for this problem is the usage of Cognitive Radio technique, which provides an intelligent management of the wide spectrum frequencies available, avoiding overburden and wasting (CABRIC; MISHRA; BRODERSEN, 2004).

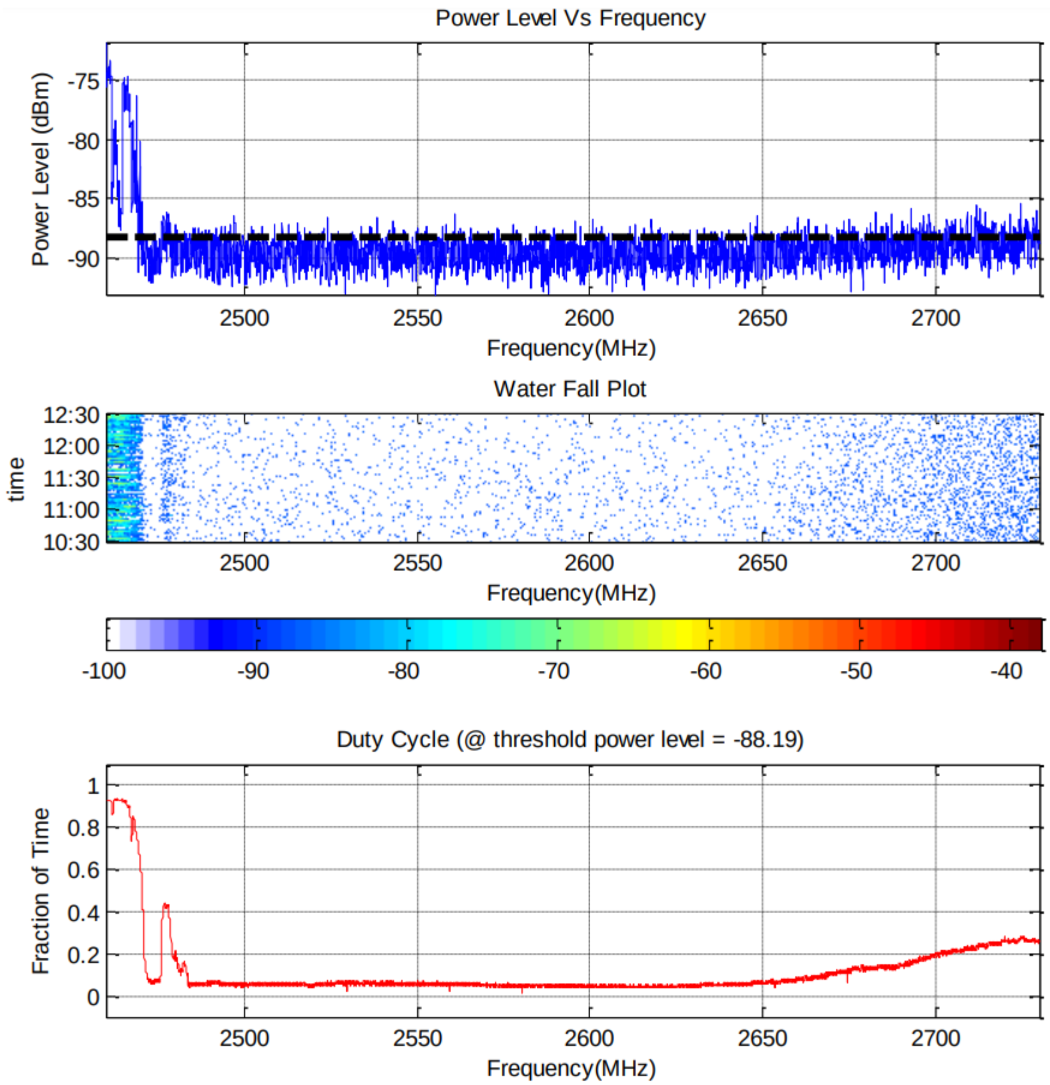
The spectrum management is executed in four steps:

- **Spectrum Sensing:** The spectrum is scanned, in steps of X Hz or MHz, searching for idle frequencies (where there is a low power gain). This information usually is passed for a master node which will choose a frequency.

- **Spectrum Decision:** In this phase one master node receives the sensing data and uses AI algorithms, like either Artificial Neural Network (ANN), Random Forest (RnF) or Hidden Markov Model (HMM), to make decisions about what frequency will be chosen to set in the next step (PINTO; CORREIA, 2018). This master node may be global or distributed in a network with different environmental contexts, as shown by (MORERIO et al., 2012) on an approach about distributed CR which presents the power of scalability of Cognitive Radio Cognitive Radio.

- **Spectrum Sharing:** After the decision, the master node transmits a message for all slaves notifying the chosen new spectrum. Since then, all nodes keep waiting for the mobility signal.

Figure 2.2 – SDR Schema



Font: Shaikh et al. (2016)

- **Spectrum Mobility:** In this phase, all nodes need to change for the new spectrum chosen. However, if this occurs without attention, some nodes could lose connection for some time, decreasing the Quality of Experience (QoE) of users in case of a non buffered stream, e.g. video call. To synchronize this there are some techniques. One simple example is the "schedule and change" which chose a exact time and every radio change to the frequency chosen at the same time.

These steps function change according to each author's approach. However, this is the main line found in the literature. In this work we will focus on Spectrum Decision phase. There are many AI algorithms utilized in related works and in the next section will be present some Machine Learning techniques that are useful for the cognitive side of CR.

GNU Radio allow the development of CR using SDR with a software radio with a Field Programmable Gate Array (FPGA) into it, in this work it is the USRP.

2.3 Software Defined Radios Tools

There are some tools to deal with Software Defined Radio (SDR) and Cognitive Radio (CR). GNU Radio was chosen between them for some features: GNU Radio is free and opensource and what most supports USRP, which is available at *Grupo de Redes Ubíquas e Comunicação (GRUBI-COM)* at *Universidade Federal de Lavras (UFLA)*.

Laboratory Virtual Instrument Engineering Workbench (LabVIEW)[1] is a programming language used to describe a virtual tool which is used in a hardware abstraction in software. Also known due its good support for USRP and easy programming mode in data flow, linking blocks, which in LabVIEW are called Virtual Instruments (VIs). The advantage of using LabVIEW is clear, however the software to develop in VIs need to be purchased. This is the great disadvantage.

Implementing Radio in Software (IRIS) was built for CR. It is also used to self reconfigurable radios in SDR maintained by Dublin University. IRIS uses Extensible Markup Language (XML) to specify the radio behaviour, changing in real time the radios settings. One application in C++ uses this XML to configure the radio environment. Moreover, IRIS uses flow graphs such as GNU Radio to describing the data flow, useful to control SDR hardware.

Open-Source SCA Implementation - Embedded (OSSIE) also its a software for SDR development. Maintained for Virginia Tech with support of National Science Foundation (NSF). Its developed in C++ offering performance and a good interface with devices using the same idea of flow graphs using block. Also it has a plugin with Eclipse IDE and GNU Radio.

ASGARD is a a platform with a flexible architecture that uses Oriented object programming (OOP) in C++. Also, set a flow and process the signal in a specific sort. The platform is divided into four elements: **Components(I)**, with the role of process the entering data; **Module(II)**, which allows the carrying of many components, responsible to set the tasks sort; **App(III)**, controls the link between components and the organization inside the module, loading this information with XML; **Communication Object(IV)**, deal with the communications between modules and components.

After analyze these tools as a whole, the best option still being GNU Radio, due its ease to load USRP and the support of Ettus Research (USRP developer company) through USRP Hardware Driver (UHD).

[1] <https://www.ni.com/pt-br/shop/labview.html>

2.4 GNU Radio

GNU Radio[2] is a free and open-source software development toolkit. Its a software signal processor, that can be used to prototype low cost RF hardware to create SDR.

There are several kind of basic blocks available for building a new software radio. Furthermore, it is possible to build a new block using the tool *gr_modtool*. This tool is used to build Out Of Tree (OOT) blocks. That is, a non default block which may use eiher Python, or C++ to be programmed. There are different types of blocks:

- The sync block allows users to write blocks that consume and produce an equal number of items per port.
- The decimation block is another type of fixed rate block where the number of input items is a fixed multiple of the number of output items.
- The interpolation block is another type of fixed rate block where the number of output items is a fixed multiple of the number of input items.
- The basic block provides no relation between the number of input items and the number of output items. All other blocks are just simplifications of the basic block. Users should choose to inherit from basic block when the other blocks are not suitable.

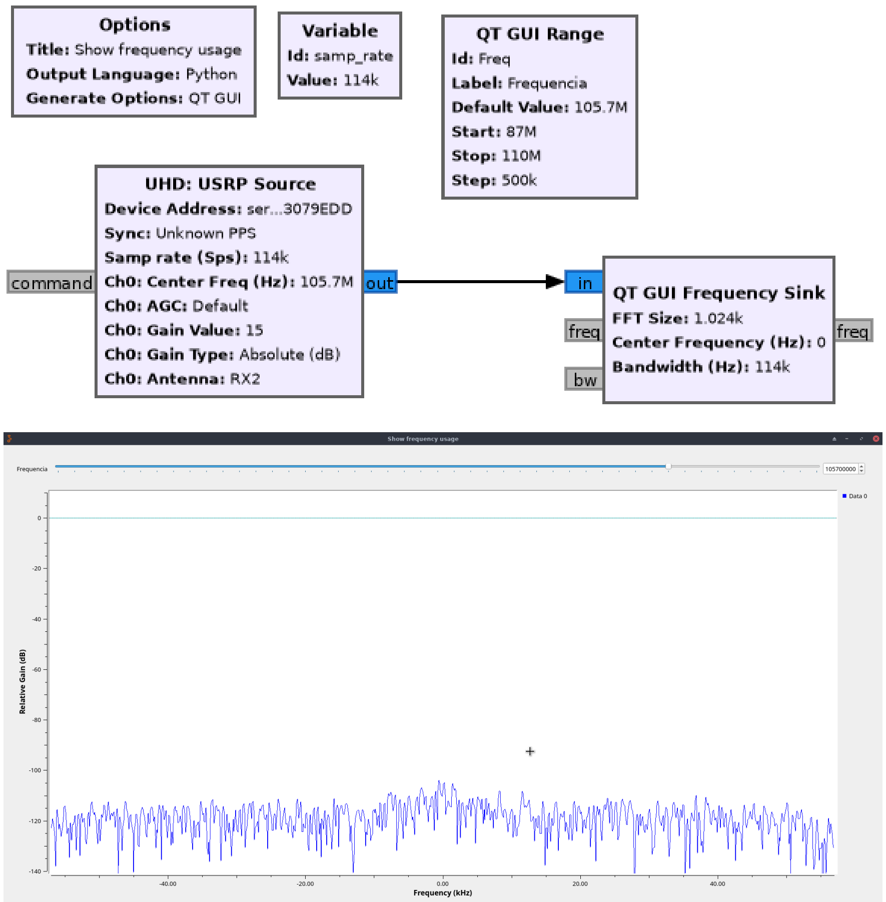
This block's behavior is related of the type of communication, that could be stream or message. In other words, stream process a constant flow of data, and message depends of events.

How seen above, some blocks has input and output of data or none of both, that is the case of basic block which usually transmit messages between blocks. The example of Figure 2.3 shows the block USRP Source plugged in a Frequency Sink, both are on default tree block. With this connection the USRP will pass its frequency data to the Sink and its will show the data using the graphic lib QT5[3].

In this dissertation, GNU Radios has an important role scanning the RF environment allowing the collection of two range of frequencies in different locations in one week. The radio used in this dissertation is the Universal Software Radio Peripheral (USRP), a software radio that has a lib called USRP Hardware Driver (UHD). UHD is a bridge between the USRP and the GNU Radio, delivering data from the spectrum such as, frequency, amplitude and power (dB).

[2] Official wiki of GNU Radio: <https://wiki.GNURadio.org/index.php/Main_Page>
[3] Official QT website: <https://www.qt.io/developers/>

Figure 2.3 – GRC example project



Font: author (2021)

2.4.1 Software Defined Radios Limitations: Timing Matters

In Bloessl et al. (2015), one of creators of GNU Radio and one important minds of SDR literature, discuss about the problem of timing in SDR. The USB protocol bounds the throughput and communication speed, due the driver implementation and USB hardware limitations.

Therefore, the idea of SDR is the implementation of radios in processor of general proposes, which one is called *software-only* solution. This architecture of software radios allows more flexibility and usually uses GNU Radio to implement its software physical layer. This approach also bring more newcomers to research in this area and allow the researcher from different areas to contribute in wireless. In the other hand, there are some impairments when using *software-only* radios. Process signal in a general propose Central Processing Unit (CPU) in a non-real-time Operating System (OS) introduces significant delays and non-determinism expressed in delay variations (SCHMID; SEKKAT; SRIVASTAVA, 2007).

One Second approach to solve this, would be the *hardware* solution, implementing the physical layer near the hardware using a FPGA, approximating the physical layer of Digital Signal Processors (DSP). Using this architecture, timings are deterministic and delay requirements of modern wireless standards can be met, however, reprogramming the system becomes complex and time consuming.

Summing up, software radios are important for the science, making easier to perform proof of concept of new architectures and protocols. However, its not possible yet to make the communications between an IEEE802.11 protocol implemented in hardware and a *software-only*. Therefore, it is an error compare bandwidth and latency with hardware radios.

2.5 Machine Learning

In this section, we will present a background about the Machine Learning (ML) algorithms used in this work.

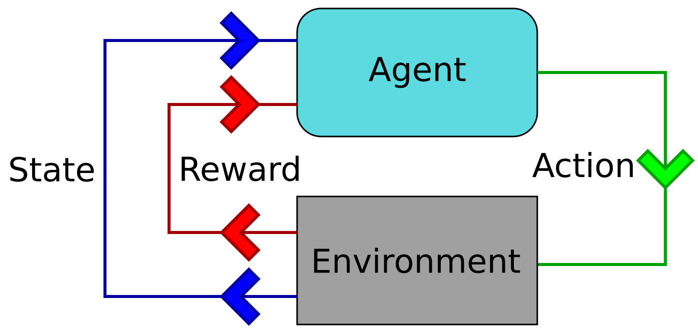
2.5.1 Reinforcement Learning

Reinforcement Learning (RL) is one of the three Machine Learning (ML) paradigms, alongside supervised learning and unsupervised learning. The RL big difference, among others, is the fact that RL does not need a previously collected dataset. RL has an agent which interact with the environment and receive rewards or punishment as feedback, and only with this information the RL is capable to learn (ANDREW, 1999).

The Figure 2.4 present the basic learning process of a RL algorithm. First, the agent makes an action over the environment, whose will react generating a reward or punishment. When the agent's action is considered wrong the environment returns a negative value, in the other hand the environment returns a positive value. When the agent finish the dataset or, in some cases, reaches some invalid state, the state reset and the agent back to the beginning, keeping only the knoledge, this is the end of the episode.

The actions are chosen aiming the maximization of the rewards received at the end of each episode. The vector S represents all possible states of the system and the vector A represents all possible actions to be taken in the environment. The agent make an action A_t at the state S_t at the time t. This action may modify the environment and the agent's state, leading him to the state S_{t+1}. Hence, the action generate, in the next time $t+1$, a reward $R_{t+1} > 0$ or a punishment $R_{t+1} < 0$. The total rewards of an episode shows the quality level of the policy adopted by the agent. This idea explains the reinforcement in the name of the technique (KAELBLING; LITTMAN; MOORE, 1996).

Figure 2.4 – RL basic workflow



Font: author (2021)

In order to improve its knowledge base, the agent must to try different combinations of the (A_t,S_t) to take knowledge of the optimal policies. Also known as the exploration process of the agent. The goal is to explore the environment and add more information to the knowledge base making some random movements. However, when the environment do not allow many mistakes, the agent can not explore the environment for a long period, so it is important to find the best exploration rate for the agent. In this case, the agent must to exploit the environment, using a previous trained data-set. Balancing exploration and exploitation is the main challenges of building RL systems.

The Reinforcement Learning field splits up in two ideas: (I) when the Markov Decision Process (MDP) is known and (II) when the MDP is unknown. When the environment is known (I) and it is possible to define statistics about it, the known MDP is the way. However, when the environment does not have a default behavior (II), or either it is not possible to predict the behaviour with high accuracy, which is the spectrum case, the best techniques to solve the spectrum is the use of the Model-Free RL (BKASSINY; LI; JAYAWEERA, 2013) such as the Q-Learning.

2.5.2 Q-Learning

Q-Learning was introduced by Watkins (1989) and it had a huge role in the RL popularization (ANDREW, 1999). Q-learning is a Model-Free RL technique, whose objective is to maximize the agent's choice policies. This characteristics makes Q-Learning ideal to improvement of AI in games.

Companies like DeepMind® uses an ensemble of techniques but with Q-learning as a basis to win a game against professionals of Dota 2 players, an online computer game based on experience and intuition with an infinity of possible movements, different of chess, which is possible to simulate with perfection all the next plays. Also, Q-Learning helps DeepMind® to build the algorithm to win games of "go", a well-known oriental difficult game to computers win because there are 10^{170} possibilities of movements since the first piece.

To update the knowledge base, the Q-Learning uses the Equation 2.1. This equation is known as Bellman equation. Like RL, the variables are the set of states S, set of actions A, and set of rewards R.

$$Q(s_t,a_t) = (1-\alpha_t)Q(s_t,a_t) + \alpha_t\left[R(s_t,a_t,s_{t+1}) + \gamma \max_a Q(s_{t+1},a)\right] \tag{2.1}$$

$\alpha \in \mathbb{R},\ 0 < \alpha < 1,\ \alpha$ is the learning rate, which the objective is to control how fast the Q-function will absolve the new knowledge. $\gamma \in \mathbb{R},\ 0 < \gamma < 1,\ \gamma$ is the discount rate. This variable determines how much important the present Q-value will have comparing with the old values. s_t is the state at the current time t and a_t is the action at the current time t. The equation is divided into two parts:

$$Q_{old} = (1-\alpha_t)Q(s_t,a_t) \tag{2.2}$$

$$Q_{new} = \alpha_t\left[R(s_t,a_t,s_{t+1}) + \gamma \max_a Q(s_{t+1},a)\right] \tag{2.3}$$

The first part, in Equation 2.2, shows the importance of the influence of the old Q-value to the new Q-value, its importance is limited by the learning rate of α.

The second part, the Equation 2.3 shows the learning rate defining the importance of the incoming value, which is the reward received at time $t+1$ plus the best discounted possible value of the next state.

The sum of both equations is the new Q-value. This will run in several episodes.

RL is important in Cognitive Radio due to its ability to change the probability over time, the data aging, but keeping certain importance level to the old data.

2.5.3 Conditional Random Fields

Conditional Random Fields (CRF) are discriminative models that can capture many correlated characteristics of the input data, allowing spatial correlations between the neighbors data, for example, words in texts or instances on time series.

The CRF is widely applied in Neural Language Processing, used to understand the grammatical function of a word in the context. Furthermore, the CRF is applied in several time-series problems, due to its ability to see in windows. Some examples are works of genetics, observing and predicting the classes of genes that were not discovered yet (LI; YUAN; WILSON, 2008) (LANNOY et al., 2011).

CRFs have a single exponential model for the joint probabilities of full paths given in the input sentence.

$$p(x,y) = \prod_t p(x_t|y_t) \sum_{j=1}^{n} p(y_t|y_{t-j}) \tag{2.4}$$

The Equation 2.4 us the relationship between the probability of the output $p(x_t|y_t)$ with the probabilities of its previous occurrences $p(y_t|y_{t-1})$. This gives CRFs the ability to forecast based on a j time window where $1 \leq j \leq n$. The hypothesis that CRFs can bring us a good result is due to this view in time window and for being a simple and fast training technique (LAFFERTY; MCCALLUM; PEREIRA, 2001).

2.6 IRace

The IRace package provides an automatic configuration tool for tuning optimization algorithms, in other words, automatically finding good configurations for the parameters values of the target algorithm saving the effort that normally requires manual tuning.

IRace is a package for the programming language R that implements a technique called Iterated Racing (LÓPEZ-IBÁNEZ et al., 2016). Iterated Racing is a method for automatic configuration of algorithms, which consists of three steps:

1. Sample new settings based on a particular distribution.

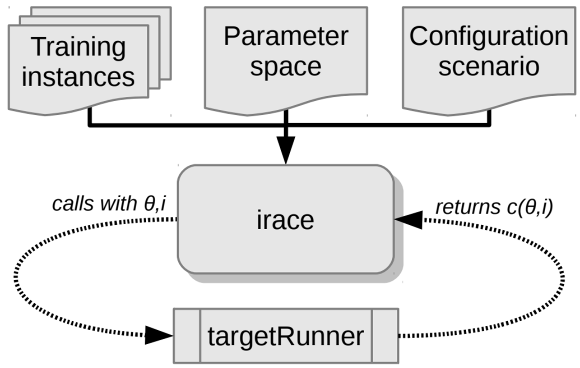
2. Select the best configuration of the newly sampled through Racing.

3. Update the sampling distribution to direct sampling to the best configuration.

These three steps are repeated until a criterion is met.

Figure 2.5 gives a general scheme of how IRace works. IRace receives as input a parameter space definition corresponding to the parameters of the target algorithm that will be tuned, a set of instances for which the parameters must be tuned for and a set of options for IRace that define the configuration scenario. Then, IRace searches in the parameter search space for good performing algorithm configurations by executing the target algorithm on different instances and with different parameter configurations. A targetRunner must be provided to execute the target algorithm with a specific parameter configuration (θ) and instance (i). The targetRunner application acts as an interface between the execution of the target algorithm and IRace: it receives the instance and configuration as arguments and must return the evaluation of the execution of the target algorithm.

Figure 2.5 – Schema IRace, flow of information



Font: López-Ibánez et al. (2016)

In IRace, each parameter has a sampling distribution associated with it. That is, independent of the distribution of other parameters, in addition to the restrictions and conditions between them. With great independence between the samples, IRace can parallelize the execution of the instances.

The stop condition in IRace is called total tuning budget. It is provides two options for setting the total tuning budget (maxExperiments and maxTime). The option maxExperiments limits the number of executions of targetRunner performed by irace. The option maxTime limits the total time of the targetRunner executions. When this latter option is used, targetRunner must return the evaluation cost together with the execution time ("cost time"). When using maxTime, irace estimates the execution time of each targetRunner execution before the configuration. The amount of budget used for the estimation is set with the option budgetEstimation (default is 2%). The obtained estimation is adjusted after each iteration using the obtained results and it is used to estimate the number of experiments that can be executed. Internally, irace uses the number of remaining experiments to adjust the number of configurations tested in each race.

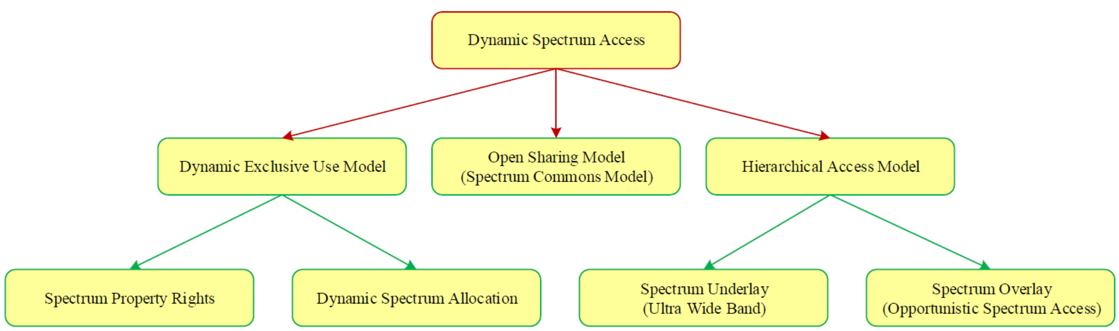
At the end of the process, a ranking is generated with individuals from the population's elite. For the experiments carried out in this work, we selected the best individual from the elite.

3 RELATED WORKS

This chapter summary the main works found in the literature about the theme of this dissertations. The research methodology includes search in digital platforms, like Google Scholar[1] IEEE Xplore Digital Library[2]. The research main string: ("cognitive radio" OR "cognitive radios" OR "mitola radios" OR "CR") AND ("decision phase" OR "artificial intelligence" OR "machine learning"). Selecting only complete articles, dissertations or thesis which approach the CR with focus on power allocation.

The research field of Dynamic Spectrum Management (DSM) is divided into paths of solutions, shown in the Figure 3.1 through the hierarchy of DSM schema. Our work approach fits into Hierarchical Access Model, more specifically, in Spectrum Overlay. First presented by Mitola, can be also regarded as opportunistic spectrum access (OSA). It finds spatial and temporal spectrum white space for SUs to use, which is also termed as the Spectrum Holes (SHs) (LIN et al., 2016).

Figure 3.1 – Dynamic Spectrum Access hierarchy



Font: Lin et al. (2016)

Several works were developed to Dynamic Spectrum Access (DSA). This section presents works that implement the decision phase of the spectrum which explore SHs. The works were divided according to the test environment in which the models were applied: test in real environments and in simulated ones.

Bkassiny, Li e Jayaweera (2013) presented an important survey for ML techniques in CR, there are many AI algorithms applied in the spectrum decision phase. However, the authors conclude that the algorithms which can adapt faster with high accuracy, have high performance predicting the spectrum of frequencies. Due to their capability to learn the behavior and readapt the model discarding gradually the old knowledge and considering the newer ones. Moreover, Bkassiny, Li e Jayaweera (2013) observe RL as one of the most promising techniques to be used in Cognitive Radios, once its behavior in continuous on time, using the MDP is relevant when thinking in the spectrum of frequencies.

[1] Accessed in 05/04/2020: <https://scholar.google.com.br/schhp?hl=pt-BR>
[2] Accessed in 05/04/2020: <https://ieeexplore.ieee.org/Xplore/home.jsp>

In (PINTO; CORREIA, 2018) were compared Artificial Neural Network (ANN), Hidden Markov Model (HMM), and Random Forest (RnF). A Cognitive Radio Network (CRN) framework was implemented using GNU Radio in order to test the proposed models in real environment using USRPs in the frequency range between 800 MHz to 5.9 GHz with 2.5 MHz intervals. The tests were performed at the same location, what could bias the results. The work raises the problem of models geographic adaptability. The decision technique chosen should be trained and tested into different locations, to compare the capability of geographic adaptability.

The paper developed by Frasch e Kwasinski (2017) do not uses techniques based in the past knowledge of the spectrum of frequencies. However, the authors consider the two primary techniques in the literature: energy detection and cyclostationary feature detection. A single-band energy detection (ED) algorithm detects the presence of a signal using a power spectral density (PSD) estimate and does not require a-priori knowledge of the noise power. A cyclostationary detection algorithm was based on (ENSERINK; COCHRAN, 1994), which uses the squared magnitude of the spectral correlation density (SCD) as a detection statistic followed by a threshold test to determine if a signal is present. The energy detection methodology proposed by the authors was implemented in a Nuand bladeRF x115 platform, which is a SDR that uses FPGA to receive the implementation of the algorithms designed with VHDL. All the tests were performed at the same place, without analyzes the adaptation.

Tumuluru, Wang e Niyato (2010) used a classical Multilayer Perceptron (MLP) to create a predictor and test its accuracy simulating the spectrum. The simulation was built with a Poisson process, to simulate the primary user traffic. Moreover, the authors warned that their model works in any kind of environment. However, simulate the spectrum of frequencies with a statistical distribution has some problems. The accuracy in represent the spectrum with a statistical model is questionable. Furthermore, the tests performed by the author were important how a proof of concept, despite the lack of tests in real environment.

Hosey et al. (2009) Implemented Q-Learning, a RL technique indicated for changing-with-time problems, in CR. In this work, the implementation is incomplete. The authors do not use the variation of Q-Learning with the greedy-ε to improve the adaptability of the model and even test the algorithm. However, the proposed implementation of dividing the spectrum in band of frequencies to building the Q-table presented an approach of Q-Learning, that were studied to build this work's approach.

GonÇalves (2019) developed a smart swap between Medium Access Control (MAC) protocols using Q-learning. When reduced this work to our spectrum decision problem, we realize that Q-learning is a interesting technique for this problem, because the spectrum decision phase also needs a continual adaptation along the CR life. In the experimental phase in FUTEBOL testbed, the Q-learning shows a high performance adapting along the tests better then the main literature works: FS-MAC and AMAC.

Table 3.1 – Related works

Work Citation	**Simulated Environment**	**Real Single Environment**	**Real Multi Environment**
Pinto e Correia (2018)		X	
Hosey et al. (2009)		X	
Tumuluru, Wang e Niyato (2010)		X	
GonÇalves (2019)		X	
Frasch e Kwasinski (2017)	X		
(ENSERINK; COCHRAN, 1994)	X		
Our approach			X

Font: author (2021)

4 METHODOLOGY

This section will present the database collection with GNU Radio in two different locations to test the algorithms. The Q-Learning and Conditional Random Fields (CRF), will be tested. And finally the methodology of test to a fair comparison between the methods.

4.1 Spectrum Range

The spectrum is a partially observable environment and it leads us to conclude that the phase of spectrum sensing is crucial to worry about how it is done. The USRP brings the values of power for each frequency range to the software. However, the hardware must choose a fragment of time in the infinite period between one sample and another. Ettus Research in the implementation of UHD do not collect the spectrum waves, but samples of it.

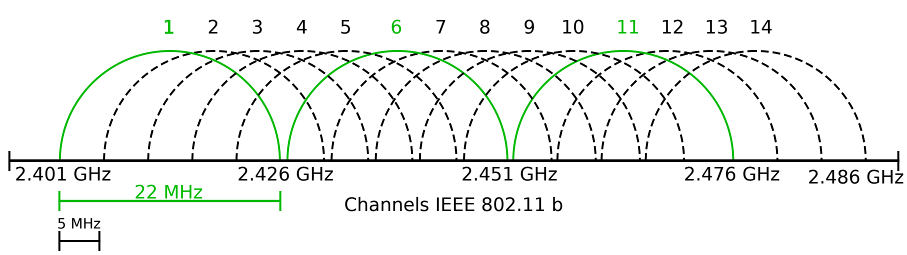
The USRP has a RX antenna that is possible use to make a full scan at the spectrum of frequencies. However it demands some time, increasing the gap between the collects. This leads us to choose a lower range for a better temporal correlation between the data. Considering this factors, was chosen a populated and common range of frequencies: the IEEE 802.11 channels 2.4 GHz and 5.8 GHz.

The Figures 4.1 and 4.2 shown the IEEE802.11 b/n channels. There are a mutual interference between the channels, due the space of 5 MHz and the channel size of 22 MHz. By ANATEL's definition, the interference between ISM bands is allowed[1].

Channel interference in a populated short range of frequencies is a spectrum environment with high concurrence, the ideal range to test predictive algorithms.

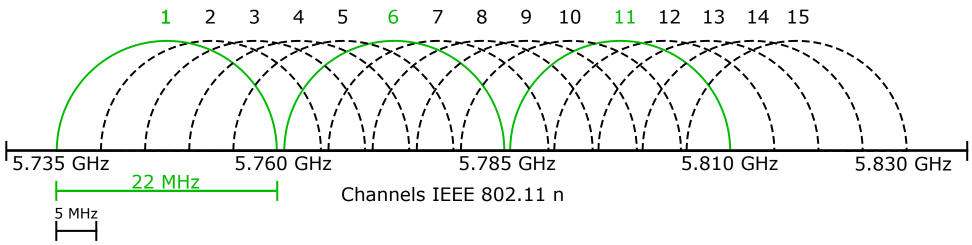
[1] Resolution Nº 716 ANATEL

Figure 4.1 – The IEEE 802.11 b channels (2.4 GHz)



Font: author (2021)

Figure 4.2 – The IEEE 802.11 n channels (5.8 GHz)



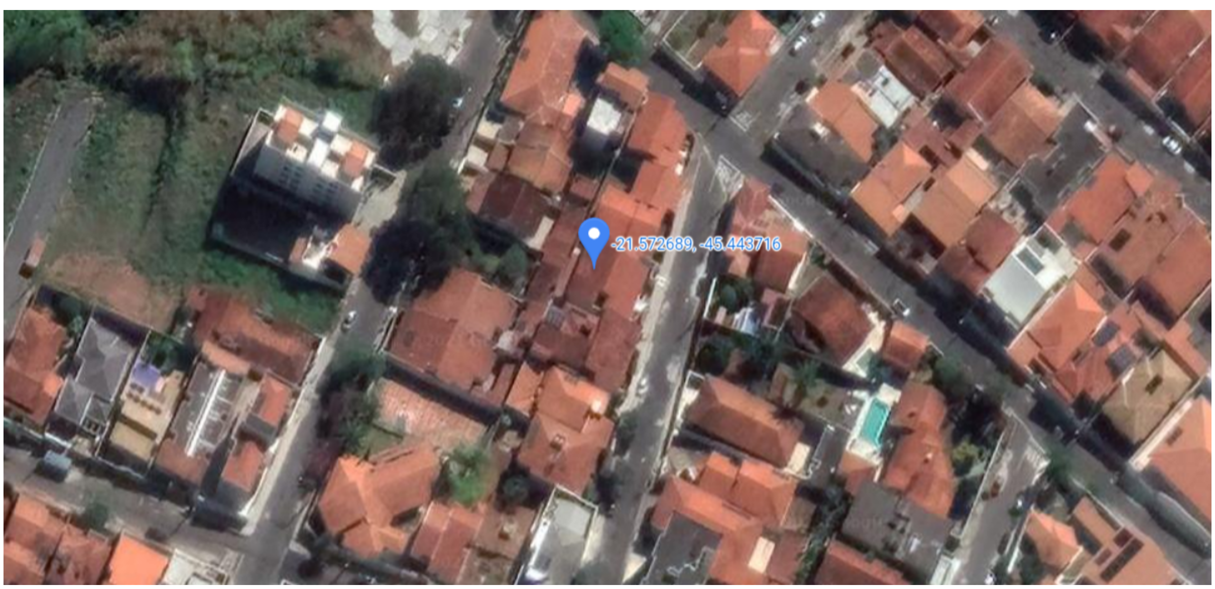
Font: author (2021)

4.2 Collection of databases

The data were collected in two different Brazilian's cities: Varginha-MG, and Pouso Alegre-MG. In Varginha the data were collected, between the days 22^{nd} and 28^{th} April, 2020, which lead to 7 days of data. In Pouso Alegre the data was collected between the 7^{th} and 13^{th} October, 2020. The date and the place are the only difference between these collects.

The Figure 4.3 and 4.4 show the exact location where the USRP were positioned to collect the data.

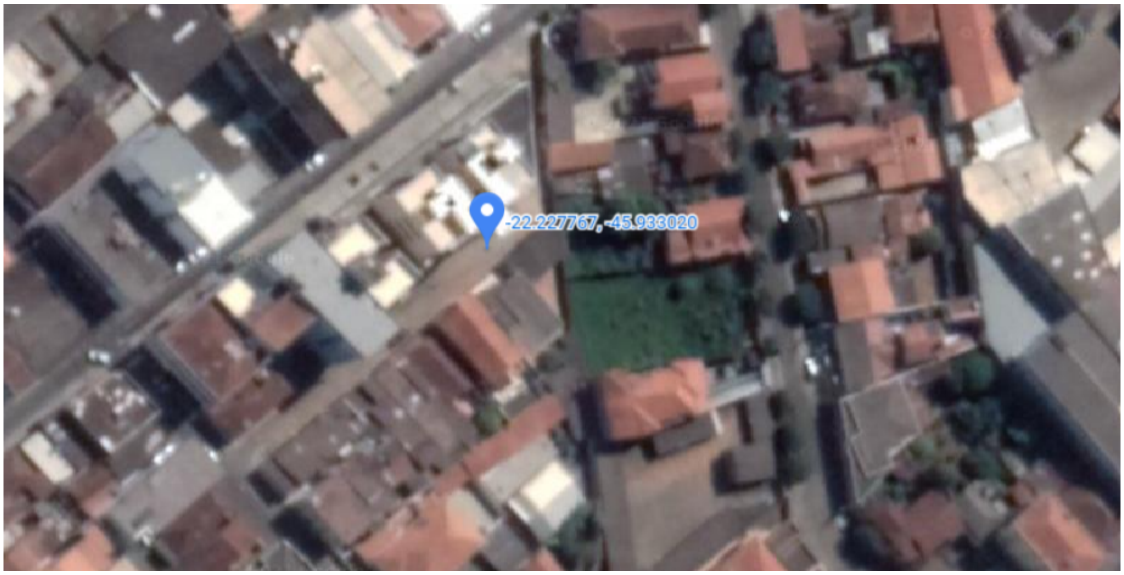
Figure 4.3 – Varginha - MG, Brazil
Cords -21,572689, -45,448716



Font: Google Maps (2021)

The Table 4.1 compares the relevant geographic factors in both locations that can interfere with the data collected. Both districts are classified as upper-middle class, which means high access to the internet. The population density is the main factor to analyze the difference between the datasets.

Figure 4.4 – Pouso Alegre - MG, Brazil
Cords -22,227767, -45,933020



Font: Google Maps (2021)

Table 4.1 – Characteristics of locations

	Varginha[2]	Pouso Alegre[3]
Population	136.603	152.549
Per capita GDP	R$40.994,76	R$51.182,28
District location	Peripheral	Central
District type	Residential	Commercial/Residential
District's predominant build	House	Building
District density	$1309 - 4875\ inhabitants/km^2$	$8263 - 13277\ inhabitants/km^2$

Font: IBGE (2010 - 2020)

The data was collected by two USRPs, one collecting 2.4 GHz, and the other 5.8 GHz. The exact interval of frequencies is 2.401 GHz - 2.495 GHz and 5.735 GHz - 5.835 GHz. The collect step is 5KHz and the USRP takes an average of time of 65 seconds to complete one scan of 2.4 GHz and 5.8 GHz. The collected features was: Date, Frequency, Power, Base Noise.

- **Date**: The exact time that the frequency was scanned;
- **Frequency**: The central frequency where the power were collected;
- **Power**: The power at the frequency;
- **Base Noise**: The Base Noise estimated power.

The collect algorithm was done in python using GNU Radio. The USRP collects the power as a function of time, to make this data understandable it is necessary to apply the algorithm of Fast Fourier Transform (FFT). Using a GNU Radio module which apply FFT to the collected data we get a FFT vector. This vector gives the power of the frequencies in mW to convert it in dBm using the Equation 4.1, which is the most usual way to present power in WiFi networks.

$$10 \times (log_{10}\left(\frac{FFT[i]}{USRP\_Rate}\right) \tag{4.1}$$

To calculate the Base Noise is used the minimal power sample collected in the central frequency, within the FFT vector. The Equation 4.2 is used to convert from mW to dBm.

$$10 \times (log_{10}\left(\frac{min(FFT)}{USRP\_Rate}\right) \tag{4.2}$$

The USRP's sample rate is the amount of samples that the USRP is capable to collect per second. The sample rate is an important parameter that need to be balanced, due the trade-off with speed of the collect and quality of the data. However, in this work the goal is to detect the presence of user, and not establish a connection with other radios. The sample rate utilized in this experiment was 10^6 sample per second. It means that to every 5KHz we divide it in 10^6 samples.

After the end of the collect, the data pass to a preprocessing step. The idea is to pass from pieces of 5KHz to a entire channel of 22 MHz, once the goal is the occupation of this channel and not the exact power per time and the labalization of this data. To make it easy we preprocess these data in three parts:

- Summarise the data of 5KHz into 5 MHz (channel piece)
- Summarise the channels pieces into 20 MHz (channel)
- Labelization

The goal here is the occupation of the channel and the electromagnetic spectrum was collected in pieces of 5KHz, so is desirable the percentage of the channel which is busy. The algorithms are present below in Algorithm 1 and 2.

Algorithm 1 Generate Channel Pieces

```
function CHANNEL_PIECES(samples)
    cont_power ← 0, cont_noise ← 0, current_piece ← 1
    i ← first frequency                        ▷ search the index of the first frequency
    channel_piece ← [ ]
    while not EOF do
        if (i − first frequency)%length of current frequency = 0 then
            current_piece = 1
        end if
        if samples[i].frequency % 5^6 = 0 then
            channel_piece[i].channel_piece ← current_piece
            channel_piece[i].power ← cont_power ÷ 1000
            channel_piece[i].noise ← cont_noise ÷ 1000
            channel_piece[i].date ← samples[i].date
            current_piece++
        end if
        if samples[i].power > −75 then
            cont_power ++
        end if
        if samples[i].noise > −90 then
            cont_noise ++
        end if
        i++
    end while
    return channel_piece
end function
```

The Algorithm 1 summarise each 5KHz power into 5 MHz, with the average of hole scan over the threshold of $-75dBm$ for power and $-90dBm$ for the noise power. The thresholds were chosen based in the safe range of confidence found in the literature. The usual interval is between $-85dBm$ to $-75dBm$ for power, and between $-100dBm$ and $-90dBm$ for noise power (CORREIA, 2006; MONKS, 2001).

The channel_piece data structure is used in the algorithm 2:

Algorithm 2 Generate Channel

function CHANNEL_PIECE_TO_CHANNEL(channel_piece)
  $channels \leftarrow [\,]$
  $scan\_size \leftarrow len(channels\_pieces)$
  $I \leftarrow 0$
  while $not\ EOF$ **do**
    if $channel\_piece.channel\_piece < scan\_size - 3$ **then**
      $channels[i].channel \leftarrow channel\_piece[i].channel\_piece$
      $channels[i].power \leftarrow max(channel\_piece[i : i+3].power)$
      $channels[i].noise \leftarrow max(channel\_piece[i : i+3].noise)$
      $channels[i].date \leftarrow channel\_piece[i].date$
      $current\_piece$++
      i++
    else
      i += 3
    end if
  end while
  return $channels$
end function

In the Algorithm 2 the data were divided into channels and scans, with the percentage of usage of each channel, now we can labeling the data-set. The label must mean if the channel in the next scan is busy or not, because the goal is to predict future behavior. In this case, a channel with 15% of usage has enough interference to be considered busy.

4.3 Metrics

To achieve an accurate and adaptive spectrum decision algorithm, we need to consider some problems and observe some metrics. The electromagnetic spectrum is more competitive in some frequency bands such as Wi-Fi (SHAIKH et al., 2016). It is also a partially observable environment, because radios has a limited capacity in collect samples from large portions of the spectrum (BKASSINY; LI; JAYAWEERA, 2013). We still having a sensitive threshold to define whether or not the SUs are interfering with the PUs (FCC, 2002).

The mobility phase of the RC spectrum is very costly (Thomas; Menon, 2017). The ideal is to reduce the radio's need to change frequency, also known as handoff rate. In order to deal with it, this work collects how many times during training and testing the model decided to change from one frequency to another, and penalized him for any change, even if it changes for free channel.

An algorithm that has been trained in a x location may not keep the accuracy in another y location, if it is not able to adapt the model to maintain accuracy (BKASSINY; LI; JAYAWEERA, 2013). In this work, is desirable maintain not only the accuracy, but also the low rate of spectrum mobility when the geographical location is changed.

In order to measure the performance of the above factors, we defined the following metrics:

- **Handoff rate**: the lower the better, it tells us if the technique has the ability to find channels that have been free longer and are less likely to change.
- **Adaptability**: the higher the better, it shows if the technique has the capacity to adapt to several different geographic environments.
- **Accuracy**: the higher the better, it shows how assertive the technique is by choosing channels that were really free in the time period $t+1$.

In the session 4.2 was presented that in both locations were collected 9,593 scans of 2.4 GHz and 9,020 scans of 5 GHz. To calculate the handoff rate and accuracy, how many times the handoffs occurred and how many times a free frequency was chosen by the algorithms. In the end, the percentage of this value in relation to the total of scans is calculated, presented below the equation of handoff rate and precision respectively.

These metrics that are collided from the algorithms applied to databases collected in different locations show the capacity of the algorithms to adapt to completely different geographical spaces.

$$handoff\_rate = \frac{\#handoffs}{\#scans} \times 100 \tag{4.3}$$

$$accuracy = \frac{\#free\ channels\ chosen}{\#scans} \times 100 \tag{4.4}$$

The adaptability equation is:

$$adaptability = \frac{\frac{\sum_{i=1}^{n} accuracy}{n} + \frac{\sum_{i=1}^{n} 1 - \overline{X}_i handoff}{n}}{2} \tag{4.5}$$

This is the generic form of the adaptability formula, which is given by the normalization of the averages of precision and handoff rate for each location. The handoff rate is subtracted from 1 as we wish to minimize it. In other words, adaptability shows how the algorithm is capable of changing the environment and maintain high accuracy and a low handoff rate. This makes adaptability our main benchmark metric to compare algorithms, since the better adaptability shows that will perform in cognitive radio.

4.4 Q-Learning

The Q-Learning can deal with environmental changes, refitting in new environments correcting the Q-Table weights and changing the policies to keep the agent's performance. This makes the Q-Learning a important candidate to be used in the decision phase of CRs.

This section present the data shaping for Q-Learning and the implementation of Q-Learning used in this work. The Algorithm 3 present an implementation with ε-greedy. This technique ensures that the agent perform a initial exploration of the environment, for better sense of the channels behaviour.

The Q-Table of CR is the Markov Process that the Q-Learning build over each iteration with updating the value based in the Bellman equation. The x and y axis of the Q-Table are the State and the Actions of the environment, in this out algorithm both line and columns are defined by the channels. A cell value in this definition is the defined by the probability to change from the channel i to channel j, how seen in Table 4.2.

Table 4.2 – Q-Table

Channels	C_0	C_i	C_n
C_0	$P(C_0 \rightarrow C_0)$	$P(C_i \rightarrow C_0)$	$P(C_n \rightarrow C_0)$
C_j	$P(C_0 \rightarrow C_j)$	$P(C_i \rightarrow C_j)$	$P(C_n \rightarrow C_j)$
C_n	$P(C_0 \rightarrow C_n)$	$P(C_i \rightarrow C_n)$	$P(C_n \rightarrow C_n)$

Font: author (2021)

The Q-Learning shown in Algorithm 3 performs in the number of the episodes which the agent will pass through the 1 week of spectrum collection. Starting "changing" from channel 1 (s) to channel 1 (a) is considered the first value of the ε-greedy to randomize if the environment will be explored or exploited. If the environment was explored is chosen a random action (channel). Otherwise, the algorithm uses the best choice from the current state (channel). Then the agent acts on the environment and the reward is gotten.

On Q-Learning for Cognitive Radios, the reward gotten is 1 if the agent chose a free channel the next time but he decides to change the channel. The reward is 4 if the chosen channel is the same as the current channel, to reduce the handoff rate. Or is -4 if the agent decides for a busy channel in the next spectrum scan.

Using the reward received from the environment and the action selected is recalculated the new Q-Value using the Bellman Equation seen at Equation 2.1.

Algorithm 3 Q-Learning: Training

function Q-LEARNING_FIT($exploration, max\_exploration, min\_exploration,$
$exploration\_decay\_rate, total\_episodes, \gamma, \alpha,$)
  $environment \leftarrow data\_set.scanns$
  $total\_scans \leftarrow environment.lenght$ ▷ # of times that each channel was collected
  $total\_channels \leftarrow max(environment[0]["channel"])$ ▷ # of channels of this frequency
  $q = float[total\_channels][total\_channels]$
  while $episode \leq total\_episodes$ **do**
    $s \leftarrow 1$
    $a \leftarrow 1$
    while $step \leq total\_scans$ **do**
      if $random(0,1) < exploration$ **then** ▷ ε-greedy
        $a = random(1,\ total\_channels)$
      else
        $a = max(q[s])$
      end if
      $r, s' = R(s, a, environment)$ ▷ gets a reward and a new state
      $q[s][a] = (1-\alpha)\ q[s][a] + \alpha(r + \gamma\ max(q[s']))$ ▷ Bellman equation
      $s = s'$
    end while
  end while
end function

On tests, is used algorithm 3. However, with only one episode to simulate real spectrum of frequencies.

4.5 Conditional Random Fields

The capability of the CRF to observe the data in space and time lead it to be chosen in this dissertation. This ability and the characteristic of the spectrum in changing with space and time, which makes the spectrum a non-stationary environment, requires an adaptable model such as the CRF.

The CRF implementation used in this work came from the Python library sklearn-crfsuite[4]. This library implements the state-of-art of CRF and it is compatible with scikit-learning[5]. The sklearn-crfsuite training input is defined by a resource dictionary. For the prediction of the electromagnetic spectrum, it is ideal to shape the input in time windows. Thus, the above measures in relation to the current are considered to predict the next. 5-time window opening has been set (the time of current input -4 time). The example below shows an entry with a window of size 2 (current time -1 time):

[4] Accessed in 13/04/2021: <https://sklearn-crfsuite.readthedocs.io/en/latest/>
[5] Accessed in 13/04/2021: <https://scikit-learn.org/stable/>

{"channel":1, "power":$P1_t$,"noise":$N1_t$,"-1:power":$P1_{t-1}$,"-1:noise":$N1_{t-1}$, ...}
{"channel":2, "power":$P2_t$,"noise":$N2_t$,"-1:power":$P2_{t-1}$,"-1:noise":$N2_{t-1}$, ...}
...
{"channel":n, "power":Pn_t,"noise":Nn_t,"-1:power":Pn_{t-1},"-1:noise":Nn_{t-1}, ...}

Since CRF is a supervised technique, it requires a vector of expected results for training. For each channel, we have a label at time t that shows that at time $t+1$ that channel will be free in the training dataset.

4.6 Test Methodology

The test methodology has the goal to collect metrics of accuracy, handoff rate and adaptabilty to analyze the AI algorithms and compare them. A fair methodology of tests was build to reach this point, all the algorithm pass throug the same process.

First the input data is processed to the format of each algorithm, once the algorithm is ready to run the the algorithms are configured to test at the IRace and have their hyper-parameters optimized. After that, the methodological process of test starts:

1. Training of the algorithm at the local x.
2. Test at the local y.
3. Collect metrics from $x \rightarrow y$.
4. Training of the algorithm at the local y.
5. Test at the local x.
6. Collect metrics from $y \rightarrow x$.
7. Gets Adaptability comparing the metrics from $x \rightarrow y$ and from $y \rightarrow x$.

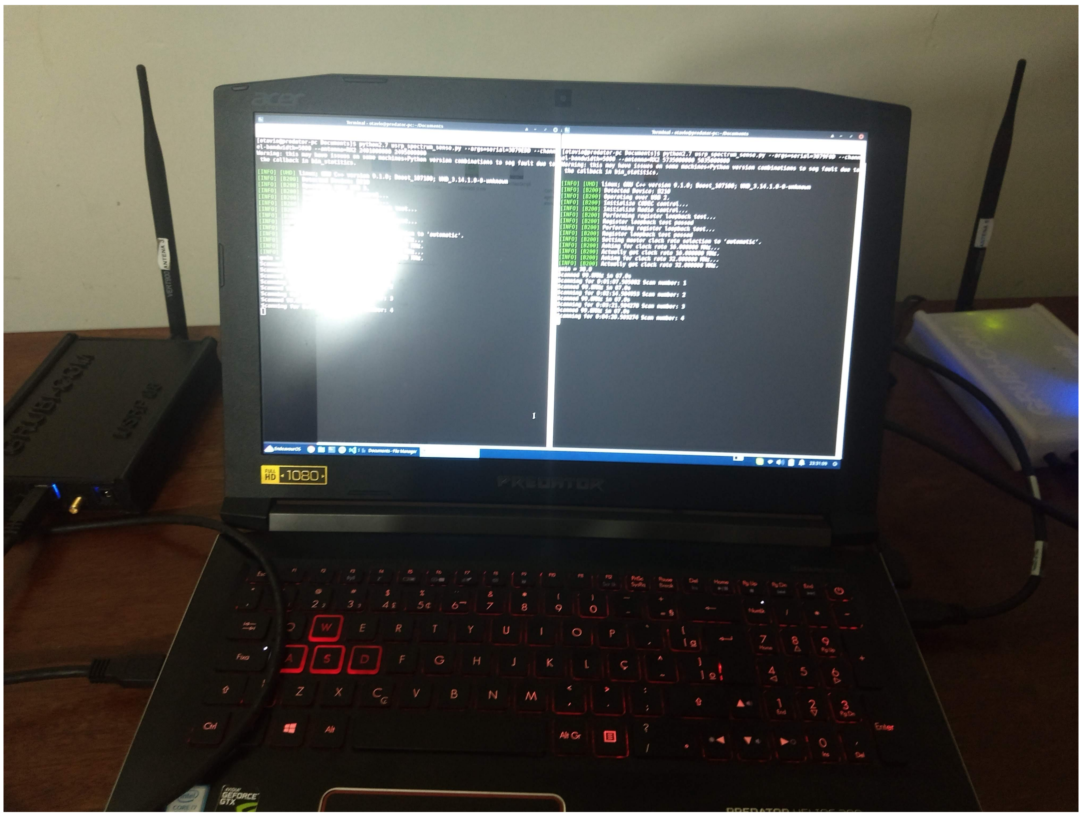
This steps are followed for each locality (Varginha and Pouso Alegre) and also for the each frequency: 2.4 GHz and 5 GHz. This methodology of tests was chosen to verify if the algorithm has some geographic dependency, in other words, analyse if the algorithm is capable to adapt in a new local without retrain the algorithm. With a high level of adaptability shows that the algorithm is capable to predict the eletromagnetic spectrum without fall in overfit in the training region.

Steps 1 to 7 are executed 100 times in each algorithm to warranty that the performance is statistically reliable. The results of acuracy and handoffs rate are given in mean, Standard Deviation (SD) and Confidence Interval (CI) with 95% of confidence.

5 RESULTS AND DISCUSSIONS

This chapter is divided into 2 sections: the first section present a data analysis about the data-set collected. The second section shows the Q-Learning and CRF results together with the discussion.

Figure 5.1 – Setup data collection



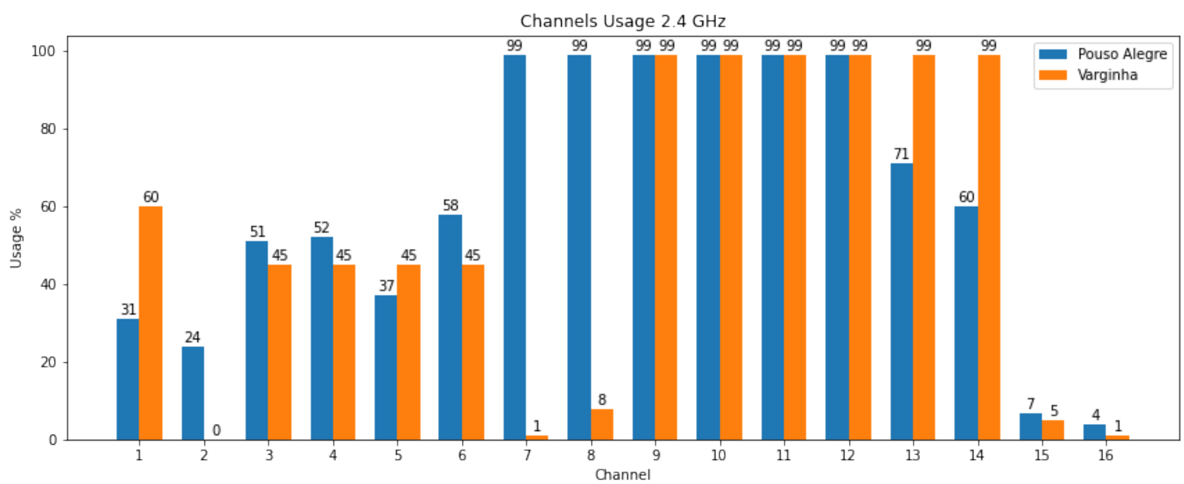
Font: author (2021)

All the data used on the tests were performed by 2 USRPs B210 with range configured in 35.5 dB connected in 1 computer. The computer is a Acer Predator Helios 300 with a Intel core i7 7700k with 16 GBs of RAM DDR4 2400 MHz.

5.1 Data Analysis

Both locations where the data was collected, Varginha and Pouso Alegre, had different features. The Table 4.1 shows the difference between the cities and the district where the USRP were positioned. These factors were important considering the results of usage analysis of the channels in the week of spectrum collection in Figures 5.2 and 5.3.

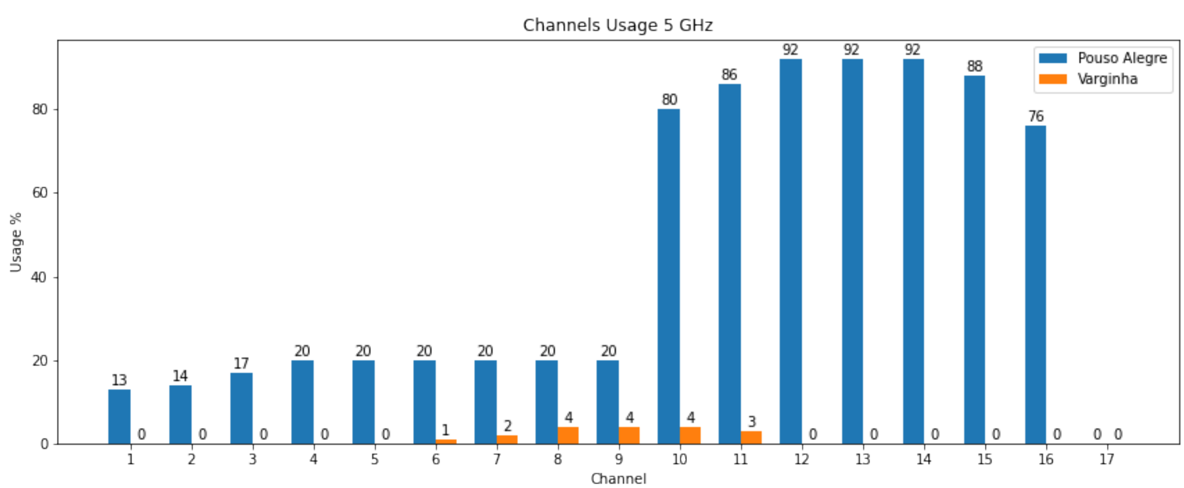
Figure 5.2 – Occupation per channel 2.4GHz



Font: author (2021)

In 2.4 GHz, channels 7-14 show high usage, reaching almost 100% of the channels occupied for an entire week in both locations, such as channels 9, 10, 11, 12. The 2.4GHz high-distance signal takes high interference between neighbors away from the collection point, even if Varginha in a population of low population density.

Figure 5.3 – Occupation per channel 5.8GHz



Font: author (2021)

In 5.8 GHz channels could be seen a high transmission power from channel 10 to 16 in Pouso Alegre. However, in Varginha the 5.8 GHz frequency band had just a few busy moments. This happens due to the vertical builds orientation in the Pouso Alegre district and a higher population density than Varginha.

Leading to more interference due to a high number of users connected in 2.4 GHz channels and nearly 90% of usage in 5GHz, which has a low-distance signal.

This analysis is crucial us to understand the results. There are desirable features in both data-sets. In 5.8GHz we have a scenario which the algorithm trained in Varginha and tested in Pouso Alegre should not perform better than trained in Pouso Alegre and tested in Varginha, this happen due the lack of information in Varginha's data-set. In 2.4 GHz we have both data-sets with high concurrence, which demands a better performance from the models.

5.2 Hyperparameters optimization

In the test methodology proposed in this work, after the data shaping to become the input of the ML algorithms, the hyperparameters of each technique were optimized using IRace. The Table 5.1 shows the hyperparameters that was chosen to be optimized and their results after pass through IRace.

The C1 and C2 parameters of the CRF is the coefficient for L1 and L2 regularization. A regression model that uses L1 regularization technique is called Lasso Regression and model which uses L2 is called Ridge Regression. The Lasso Regression adds absolute value of magnitude of coefficient as penalty term to the loss function and the Ridge Regression adds squared magnitude of coefficient as penalty term to the loss function.

Table 5.1 – Results IRace CRF

Algorithm	**c1**	**c2**	**max_iterations**	**all_possible_transitions**
lbfgs	0,501	0,3644	6	False

Font: author (2021)

Table 5.2 presents the hyperparameterization of Q-Learning with ε-gready strategy, used in this work. The values of γ and α are, respectively, the discount rate and the learning rate of the Bellman equation. The IRace also optimized the number of episodes to improve adjust the other hyperparameters to train faster. Other values are related to ε behavior. In tests, the exploration rate is reduced and the exploration attributes set to zero since the total episodes are set to one. The reason for this is to reproduce the real environment, where the agent can not "play the game" of exploring the spectrum holes more than once.

Table 5.2 – Results IRace Q-learning

	exploration (ε)	**exploration max**	**exploration min**	**exploration decay rate**	γ	α	**total episodes**
Train	0.5792	0.5792	0.4979	0.01	0.6	0.53	991

Font: author (2021)

Section 2.6 shows that the IRace's role is the minimization of the objective function, in this case the CRF and Q-Learning results. The goal of the algorithms is to maximize the accuracy and minimize the handoff rate, so the output of the algorithm received as input in IRace were defined by equation 5.1.

$$minimize(x) = (accuracy + (1 - handoff\_rate)) \times -1 \tag{5.1}$$

This is the minimization function to correlate the accuracy and handoff rate. The IRace sees the target's algorithm result as a cost, so the main goal is to minimize the problem, and the accuracy is positive for the spectrum decision problem and the handoff rate is negative. This explains the choice of the parameters.

5.3 Algorithms results

After both models were applied in the tests methodology described in section 4.6, the results were collected and presented in the Tables 5.3 and 5.4. The results were divided into the two Wi-Fi frequency bands, due to the different features of 2.4GHz and 5.8GHz. A radio connected in the frequencies 5.8GHz reaches a lower distance than a connected in 2.4GHz.

Table 5.3 shows the 2.4GHz results. CRF gets 98.47% of adaptability and Q-Learning 92.29%. The hypothesis about what motivates this difference in the CRF capability of temporal analysis due to its time window. On the other hand, the Q-Learning has an immediate observation of the current time slot, this bounds the algorithm with less information to make correlations and take better decisions.

Furthermore, the Q-Learning gets 9.62% in average of handoff rate between the two scenarios of 2.4GHz, which is a high value when put it against the CRF handoff results which gets 0,013% in average. The accuracy of both were statistically equal in the scenario of training in Pouso Alegre and test in Varginha, since this Pouso Alegre has a more busy spectrum than Varginha. However, in the other scenario, testing in Varginha and training in Pouso Alegre, both algorithms did not perform well, with 93.05% accuracy of Q-Learning and 95.78% from CRF.

Table 5.3 – Results 2.4 GHz - Varginha - Pouso Alegre

Algorithm	**Metric**	**Train**	**Test**	**SD**	$\overline{X}\pm$ **CI (95%)**
Q-Learning	Accuracy	Varginha	Pouso Alegre	0,65%	93,05% ± 0,12%
		Pouso Alegre	Varginha	0,57%	98,13% ± 0,11%
	Handoff	Varginha	Pouso Alegre	1,14%	9,64% ± 0,24%
		Pouso Alegre	Varginha	2,45%	9,60% ± 0,12%
	Adapt.	-	-	-	92,29%
CRF	Accuracy	Varginha	Pouso Alegre	0.0%	95,78% ± 0,000%
		Pouso Alegre	Varginha	0,018%	98,15% ± 0,003%
	Handoff	Varginha	Pouso Alegre	0,018%	0,028% ± 0,003%
		Pouso Alegre	Varginha	0,009%	0,026% ± 0,002%
	Adapt.	-	-	-	98,47%

Font: author (2021)

Table 5.4 shows the 5.8GHz results. In this frequencies bands, the Q-Learning gets 99.08% of adaptability and CRF 92.29%. The interference in 5.8GHz is lower, due the low transmission distance. How seen in Figure 5.3, in Varginha the usage of the channels is was less then 4% and in Pouso Alegre the average of usage was 48,12%. This makes Varginha an environment with lack of information for the 5.8GHz frequencies.

The CRF gets a unstable accuracy, due the training at Varginha its gets an average of 87,06% in tests in Pouso Alegre. Differently, when training in Pouso Alegre and testing in Varginha, the result was 100% of accuracy. The Q-Learning gets an stable result with average of 99,24% in both scenarios, however the Q-Learning still with a high handoff rate when comparing with CRF, 1,98% against 0%.

Table 5.4 – Results 5.8 GHz - Varginha - Pouso Alegre

Algorithm	**Metric**	**Train**	**Test**	**SD**	$\overline{X}\pm$ **CI (95%)**
Q-Learning	Accuracy	Varginha	Pouso Alegre	0.152%	98.51% ± 0.105%
		Pouso Alegre	Varginha	0.033%	99.97% ± 0,007%
	Handoff	Varginha	Pouso Alegre	0,172%	1,98% ± 0,120%
		Pouso Alegre	Varginha	0,075%	1.98% ± 0,014%
	Adapt.	-	-	-	99,08%
CRF	Accuracy	Varginha	Pouso Alegre	0,00%	87,06% ± 0,00%
		Pouso Alegre	Varginha	0,00%	100,00% ± 0,00%
	Handoff	Varginha	Pouso Alegre	0,00%	0,00% ± 0,000%
		Pouso Alegre	Varginha	0,00%	0,00% ± 0,000%
	Adapt.	-	-	-	96,76%

Font: author (2021)

To finish the experiments, the 2 cities data-sets were joined into one, and divided into 2 scenarios. One scenario with 75% of the data-set used for training and 25% for tests, the other scenario with 25% for train and 75% for tests. On both scenarios, the time flow were considered. This tests allow to analyse the difference that the amount of training and test influence on the results. To improve the understandably, lets use the scenario with 75% of training as high-trained and the scenario with 25% as low-trained.

Table 5.5 – Results 2.4 GHz - Dataset Joined

Algorithm	**Metric**	**Train**	**Test**	**SD**	$\overline{X}\pm$ **CI (95%)**
Q-Learning	Accuracy	75%	25%	0.531%	90.72% ± 0,104%
		25%	75%	0.542%	95.70% ± 0.106%
	Handoff	75%	25%	8.717%	11.497% ± 1.708%
		25%	75%	7.521%	7.116% ± 1.4%
	Adapt.	-	-	-	91.9515%
CRF	Accuracy	75%	25%	0.028%	91.991% ± 0.005%
		25%	75%	0.04%	96.659% ± 0.007%
	Handoff	75%	25%	0.048%	0.064% ± 0.009%
		25%	75%	0.009%	0.021% ± 0.001%
	Adapt.	-	-	-	97.124%

Font: author (2021)

At table 5.5 could be seen that the variation of the handoff rate of the Q-Learning is grater than the CRF's. The same behaviour found in the 2.4 GHz at table 5.3 and more than 4 times greater.

Observing the accuracy of the Q-Learning we can notice that both stay over the 90%. However the handoff rate with 11% for high-trained and 7% for the low trained. This makes the Q-Learning not recommended to deal with this types of scenarios in a high concurrent environment such 2.4GHz.

Even with the low handoff rate for the CRF, the both model shown in these 2 scenarios in 2.4 GHz a non-recommended for use in a commercial . The CRF got the better performance, with 97% of adaptability against 92% of Q-Learning.

In both models with high-trained data-set got a worst accuracy comparing with the low-trained. This is the consequence of overfitting, the high-trained uses way more training data than the cross tests over cities, this is why this tests were not included on the methodology of tests. However it is important present the results to justify.

Table 5.6 – Results 5.8 GHz - Dataset Joined

Algorithm	Metric	Train	Test	SD	$\overline{X}\pm$ CI (95%)
Q-Learning	Accuracy	75%	25%	0.058%	99.864% $\pm$ 0,011%
		25%	75%	0.306%	98.555% $\pm$ 0.060%
	Handoff	75%	25%	0.208%	0.727% $\pm$ 0.040%
		25%	75%	0.450%	2.216% $\pm$ 0.088%
	Adapt.	-	-	-	98.869%
CRF	Accuracy	75%	25%	0.0 %	99.911% $\pm$ 0.0%
		25%	75%	0.0%	91.286% $\pm$ 0.0%
	Handoff	75%	25%	0.0%	0.02% $\pm$ 0.0%
		25%	75%	0.0%	0.0% $\pm$ 0.0%
	Adapt.	-	-	-	97.794%

Font: author (2021)

Comparing with 2.4 GHz, the tests in 5 GHz frequencies, in these scenarios, showed more viable applicability of the Q-Learning. The model performs over 98.2% considering the standard deviation lower bound. The handoff rate got lower than 2.7% on the low-trained scenario.

In 5.8 GHz, the CRF keeps its low handoff rate capability, however got a worse accuracy in the low-trained scenario, comparing with Q-Learning.

The results shown the viability of the models in some scenarios. However, it is necessary a bigger base of knowledge, in more locations, to achieve a with more confidence that the adaptability of the model and achieve a more viability of the Cognitive Radios in a commercial way.

6 CONCLUSION

The use of Industrial, Scientific & Medical (ISM) bands is growing each year, increasing the interference between devices. To avoid this, radios need to be aware of the wireless environment and reconfigure their parameters with high precision. The Cognitive Radio (CR) connects in the spectrum of frequencies opportunistically, which allows Secondary Users (SUs) to connect in Primary Users (PUs) frequencies without break the exclusive use contract. Using SDR, tests in the real spectrum of frequencies become simpler with USRPs and GNURadio. In this work, we proposed a methodological and standardized set of tests to evaluate CR decision phase algorithms. Differently from the literature works, was considered not only the accuracy, but also the reduction of handoff rate and the model adaptability in different geographic locations.

The chosen algorithms for this work were Q-Learning and Conditional Random Fields (CRF). The goal is test the algorithms in two locations and compare the capability of geographic adaptation of each technique. Different from other literature works, with this methodology of tests and the algorithms aiming minimize the handoff rate and improve the adaptability of the models, the Cognitive Radio (CR) might increase mobility and reduce handoff rate. The CRF results in the proposed test methodology shown a better consistency with 97.6% of adaptability average in both location and less than 1% of handoff rate with 95% of confidence. In the other hand, Q-Learning shown a better consistency in 5.8GHz data-sets, which has the challenge of low level of information from Varginha in 5.8GHz.

Q-Learning got consistent results due to its continual learning during the tests. However, the Q-Learning handoff rate is higher than CRF due to its exploration rate. These factors make of Q-Learning only with $\varepsilon - gready$ not ideal for predicting the spectrum. The CRF with a low handoff rate and high adaptability in more concurrent environments shows that CRF might be applied in CR in urban centers with high spectrum usage.

The high-trained and low-trained experiments shown the overfitting of the models when joining the 2 cities and selecting 75% of the data-set to train. This might shown that the models need to reduce the memory to forget the old data, even in the training, because larger data-set could be used.

Using our approach of tests standardization with at least two or more locations, to compare the capability of the algorithms on adapting in RF environment, we propose more analysis. In future works, is relevant to test other literature algorithms, that were not tested in more than one real RF environment, and observe is the accuracy and handoffs rate are preserved.

Another valid future work, now based on the Q-Learning results, is other implementation of Q-Learning with Upper Confidence Bound (UCB) exploration. Using a different exploration technique might improve the adaptability, since this technique is more adaptive in some scenarios Markov Decision Process

(MDP). Other possible technique for a future work that can achieve good adaptability is Deep Reinforcement Learning (DRL) with Deep Q-Learning, although it presents problems in dealing with the non-stationary nature of the electromagnetic spectrum, this techniques allow a thorough modeling of the problem. For example, is possible to define more labels, input with more spectrum variables, since the Deep Q-Learning has a better performance with a larger amount of input data.

REFERENCES

AKYILDIZ, I. F. et al. A survey on spectrum management in cognitive radio networks. **IEEE Communications magazine**, IEEE, v. 46, n. 4, 2008.

ANDREW, A. M. Reinforcement learning: An introduction by richard s. sutton and andrew g. barto, adaptive computation and machine learning series, mit press (bradford book), cambridge, mass., 1998, xviii+ 322 pp, isbn 0-262-19398-1. **Robotica**, Cambridge University Press, v. 17, n. 2, p. 229–235, 1999.

BKASSINY, M.; LI, Y.; JAYAWEERA, S. K. A survey on machine-learning techniques in cognitive radios. **IEEE Communications Surveys & Tutorials**, IEEE, v. 15, n. 3, p. 1136–1159, 2013.

BLOESSL, B. et al. Timings matter: Standard compliant ieee 802.11 channel access for a fully software-based sdr architecture. **ACM SIGMOBILE Mobile Computing and Communications Review**, ACM, v. 18, n. 3, p. 81–90, 2015.

CABRIC, D.; MISHRA, S. M.; BRODERSEN, R. W. Implementation issues in spectrum sensing for cognitive radios. In: IEEE. **Signals, systems and computers, 2004. Conference record of the thirty-eighth Asilomar conference on**. [S.l.], 2004. v. 1, p. 772–776.

CHOWDHURY, K. R.; AKYILDIZ, I. F. Cognitive wireless mesh networks with dynamic spectrum access. **IEEE Journal on Selected Areas in Communications**, v. 26, n. 1, p. 168–181, 2008.

CORREIA, L. H. A. **Técnicas de controle de potência para protocolos MAC em redes de sensores sem fio**. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2006.

ENSERINK, S.; COCHRAN, D. A cyclostationary feature detector. In: IEEE. **Proceedings of 1994 28th asilomar conference on signals, systems and computers**. [S.l.], 1994. v. 2, p. 806–810.

FCC. Spectrum policy task force releases report setting forth recommendations for spectrum policy reform. **Report of Spectrum Policy Task Force (SPTF)**, v. 4, 2002.

FRASCH, I.; KWASINSKI, A. Wideband spectrum holes detection implementation for cognitive radios. In: IEEE. **2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. [S.l.], 2017. p. 278–282.

GONÇALVES, A. V. G. S. **Troca automática de protocolos MAC empregando aprendizado por reforço**. Dissertação (Mestrado) — UFMG - Universidade Federal de Minas Gerais, 2019.

HOSEY, N. et al. Q-learning for cognitive radios. In: NATIONAL UNIVERSITY OF IRELAND MAYNOOTH. **Proceedings of the China-Ireland Information and Communications Technology Conference (CIICT 2009). ISBN 9780901519672**. [S.l.], 2009.

HUANG, N. E. et al. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. **Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences**, The Royal Society, v. 454, n. 1971, p. 903–995, 1998.

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. **Journal of artificial intelligence research**, v. 4, p. 237–285, 1996.

KOLODZY, P.; AVOIDANCE, I. Spectrum policy task force. **Federal Commun. Comm., Washington, DC, Rep. ET Docket**, v. 40, n. 4, p. 147–158, 2002.

LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. v. 18, p. 282–289, 2001.

LANNOY, G. D. et al. Weighted conditional random fields for supervised interpatient heartbeat classification. **IEEE Transactions on Biomedical Engineering**, IEEE, v. 59, n. 1, p. 241–247, 2011.

LI, C.-T.; YUAN, Y.; WILSON, R. An unsupervised conditional random fields approach for clustering gene expression time series. **Bioinformatics**, Oxford University Press, v. 24, n. 21, p. 2467–2473, 2008.

LIN, Y. et al. A novel dynamic spectrum access framework based on reinforcement learning for cognitive radio sensor networks. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 10, p. 1675, 2016.

LÓPEZ-IBÁNEZ, M. et al. The irace package: Iterated racing for automatic algorithm configuration. **Operations Research Perspectives**, Elsevier, v. 3, p. 43–58, 2016.

MARQUES, A. F. et al. Sirco-um protocolo de sensoriamento inteligente para rádios cognitivos. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2016.

MIN, A. W. et al. Opportunistic spectrum access for mobile cognitive radios. In: IEEE. **2011 Proceedings IEEE INFOCOM**. [S.l.], 2011. p. 2993–3001.

MITOLA, J. Software radios. **IEEE Communications magazine**, v. 33, n. 5, p. 24–25, 1995.

MITOLA, J.; MAGUIRE, G. Q. Cognitive radio: making software radios more personal. **IEEE Personal Communications**, v. 6, n. 4, p. 13–18, Aug 1999. ISSN 1070-9916.

MONKS, J. P. **Transmission power control for enhancing the performance of wireless packet data networks**. Tese (Doutorado) — University of Illinois, 2001.

MORERIO, P. et al. Distributed cognitive radio architecture with automatic frequency switching. In: **2012 Complexity in Engineering (COMPENG). Proceedings**. [S.l.: s.n.], 2012. p. 1–4.

PINTO, L. R. M.; CORREIA, L. H. A. Analysis of machine learning algorithms for spectrum decision in cognitive radios. In: IEEE. **2018 15th International Symposium on Wireless Communication Systems (ISWCS)**. [S.l.], 2018. p. 1–6.

RONDEAU, T. W. et al. Cognitive radios with genetic algorithms: Intelligent control of software defined radios. In: **Software defined radio forum technical conference**. [S.l.: s.n.], 2004. p. C3–C8.

SCHMID, T.; SEKKAT, O.; SRIVASTAVA, M. B. An experimental study of network performance impact of increased latency in software defined radios. In: ACM. **Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization**. [S.l.], 2007. p. 59–66.

SHAIKH, B. et al. Worldwide spectrum sensing measurements and the way forward for cognitive radio networks: A survey. **Asian Journal of Engineering, Sciences & Technology**, 2016.

Thomas, J.; Menon, P. P. A survey on spectrum handoff in cognitive radio networks. In: **2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)**. [S.l.: s.n.], 2017. p. 1–4.

TUMULURU, V. K.; WANG, P.; NIYATO, D. A neural network based spectrum prediction scheme for cognitive radio. In: IEEE. **2010 IEEE International Conference on Communications**. [S.l.], 2010. p. 1–5.

WATKINS, C. J. C. H. **Learning from delayed rewards**. Tese (Doutorado) — King's College, Cambridge United Kingdom, 1989.

WELLENS, M.; RIIHIJARVI, J.; MAHONEN, P. Evaluation of adaptive mac-layer sensing in realistic spectrum occupancy scenarios. In: IEEE. **2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)**. [S.l.], 2010. p. 1–12.

WU, C. et al. Spectrum management of cognitive radio using multi-agent reinforcement learning. In: INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. **Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Industry track**. [S.l.], 2010. p. 1705–1712.

XING, X. et al. Channel quality prediction based on bayesian inference in cognitive radio networks. In: IEEE. **2013 Proceedings IEEE INFOCOM**. [S.l.], 2013. p. 1465–1473.