



LUCCAS RAFAEL MARTINS PINTO

**DECISÃO DO ESPECTRO BASEADO EM APRENDIZADO DE
MÁQUINA PARA REDES DE RÁDIOS COGNITIVOS COM
SUPORTE A COMUNICAÇÃO MULTI-SALTOS**

LAVRAS – MG

2018

LUCCAS RAFAEL MARTINS PINTO

**DECISÃO DO ESPECTRO BASEADO EM APRENDIZADO DE MÁQUINA PARA
REDES DE RÁDIOS COGNITIVOS COM SUPORTE A COMUNICAÇÃO
MULTI-SALTOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

Prof. Dr. Luiz Henrique Andrade Correia
Orientador

**LAVRAS – MG
2018**

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

Martins Pinto, Luccas Rafael

Decisão do Espectro baseado em Aprendizado de Máquina para Redes de Rádios Cognitivos com suporte a comunicação multi-saltos / Luccas Rafael Martins Pinto. 1^a ed. rev., atual. e ampl. – Lavras : UFLA, 2018.

95 p. : il.

Dissertação(mestrado acadêmico)–Universidade Federal de Lavras, 2018.

Orientador: Prof. Dr. Luiz Henrique Andrade Correia.

Bibliografia.

1. Rádios Cognitivos. 2. Aprendizado Máquina. 3.*Hidden Markov Model*. I. Universidade Federal de Lavras. II. Título.

LUCCAS RAFAEL MARTINS PINTO

**DECISÃO DO ESPECTRO BASEADO EM APRENDIZADO DE MÁQUINA PARA
REDES DE RÁDIOS COGNITIVOS COM SUPORTE A COMUNICAÇÃO
MULTI-SALTOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

APROVADA em 25 de Outubro de 2018.

Prof. Dr. Daniel Fernandes Macedo UFMG
Prof. Dr. Ahmed Ali Abdalla Esmin UFLA

Prof. Dr. Luiz Henrique Andrade Correia
Orientador

**LAVRAS – MG
2018**

RESUMO

Os avanços tecnológicos nos últimos anos reduziram os custos de fabricação e distribuição de dispositivos sem fio, contribuindo com o aumentando o número e a quantidade de aplicações que utilizam esses dispositivos, sendo que grande parte dessas aplicações utilizam as frequências chamadas ISM (*Industrial, Scientific, and Medical*). Por este motivo, essas faixas de frequência encontram-se atualmente sobrecarregadas e assim a ocupação do espectro está sujeita a sofrer problemas de sobreposição, dado o crescimento do volume de dispositivos de comunicação sem fio.

Com o objetivo de reduzir a interferência, causada pela grande utilização de faixas do espectro, que gera perdas de informação e diminuição do desempenho de uma rede, os Rádios Cognitivos emergem como uma solução inteligente para esse problema de ocupação do espectro. Os Rádios Cognitivos são rádios definidos por *software*, que são capazes de realizar de maneira autônoma a reconfiguração da rede por meio de aprendizado e adaptação do meio em que se encontra inserido. Por esse motivo, diversas pesquisas foram realizadas para o desenvolvimento de *frameworks* que fossem utilizados na implementação de Redes de Rádios Cognitivos. Contudo, a maior parte das pesquisas desenvolveram e avaliaram esses *frameworks* utilizando simuladores de redes sem fio. Apesar desses simuladores refletirem o ambiente de uma rede sem fio, as características dinâmicas do espectro não são corretamente modeladas no contexto dos Rádios Cognitivos. Assim, este trabalho propõe a extensão de uma arquitetura para Redes de Rádios Cognitivos que visa o desenvolvimento de métodos para a decisão do espectro baseados em algoritmos de aprendizado de máquina, além de oferecer a implementação de protocolos de roteamento para a camada de rede, de modo a permitir a comunicação multi-hop.

O *framework* desenvolvido contou com a utilização de dois métodos para decisão do espectro baseados em aprendizado de máquina, sendo que o primeiro foi o *Random Forest*, que é baseado em árvores de decisão, e o segundo foi o *Hidden Markov Model*, um processo estocástico que utiliza a propriedade de Markov na previsão estados futuros não observáveis. Além disso, os protocolos desenvolvidos para a comunicação multi-hop foram o *Ad Hoc On-Demand Distance Vector* um protocolo reativo, e o *Optimized Link State Routing* um protocolo pró-ativo.

Os resultados obtidos mostraram que os métodos de aprendizado de máquina foram capazes de escolher faixas de frequência do espectro de modo a diminuir a coexistência e assim melhorar o desempenho da rede. O método que se destaca nesse ponto foi o *Hidden Markov Model* que obteve melhores resultados em relação a taxa de entrega de pacotes.

Palavras-chave: Rádio Cognitivo; Aprendizado de Máquina; *Hidden Markov Model*; *Random Forest*

ABSTRACT

The technological advances in recent years have reduced the costs of manufacturing wireless devices, increasing the number of applications that use these devices, and most of these applications use the so-called ISM (Industrial, Scientific, and Medical). Therefore, these frequency bands are currently overloaded and thus spectrum occupancy is subject to overlapping problems given the growth in the volume of wireless devices.

In order to reduce the interference caused by the large use of spectrum bands that generate information loss and decrease network performance, Cognitive Radios emerge as a smart solution to this problem of spectrum occupancy. Cognitive Radios are software-Defined Radios, which are capable of autonomously performing the reconfiguration of the network by learning and adapting the medium in which it is inserted. For this reason, several types of research were carried out to develop frameworks that could be used in the implementation of Cognitive Radio Networks. However, most researches have developed and evaluated these frameworks using wireless simulators. Although these simulators reflect the environment of a wireless network, the dynamic characteristics of the spectrum are not correctly modeled in the context of Cognitive Radios. Thus, this work proposes the extension of an architecture for Cognitive Radio Networks that aims at the development of methods for spectrum decision based on machine learning algorithms, besides offering the implementation of routing protocols for the network layer to allow multi-hop communication.

The developed framework relied on the use of two methods based on machine learning, the first one was the Random Forest that is based on decision trees and the second was the Hidden Markov Model, one stochastic process that uses the Markov property in predicting future unobservable states. In addition, the protocols developed for multi-hop communication were the Ad Hoc On-Demand Distance Vector a reactive protocol, and Optimized Link State Routing a proactive protocol.

The results showed that the machine learning methods were able to choose frequency bands of the spectrum in order to decrease the coexistence and thus improve the performance of the network. The method that stands out at this point was the Hidden Markov Model that obtained better results in the package delivery rate.

Keywords: Cognitive Radio; Machine Learning; Hidden Markov Model; Random Forest

LISTA DE FIGURAS

Figura 2.1 – Arquitetura de uma rede de rádios cognitivos baseado em Akyildiz et al. (2008)	18
Figura 2.2 – Mapa dos equipamentos implantados DCC/UFMG (FUTEBOL, 2018)	33
Figura 3.1 – Arquitetura do protocolo SIRCo Marques (2016).	37
Figura 3.2 – Fluxo de mensagens SIRCo Marques (2016).	39
Figura 4.1 – Arquitetura do <i>framework</i> adaptado de Souza et al. (2017)	40
Figura 4.2 – Fluxo de Mensagens entre 3 nós	43
Figura 5.1 – Árvore de decisão adaptado de Quinlan (1986)	45
Figura 5.2 – Exemplo Simplificado de uma <i>Random Forest</i>	46
Figura 5.3 – Perceptron de uma camada adaptado de Rosenblatt (1958)	56
Figura 5.4 – Rede MLP com duas camadas intermediárias	57
Figura 6.1 – Formato do pacote de mensagens de controle	61
Figura 7.1 – Exemplo de uma árvore retirada da floresta	68
Figura 7.2 – Exemplo de uma árvore reduzida	69
Figura 7.3 – Fluxograma do <i>Random Forest</i>	69
Figura 7.4 – Modelagem para <i>Hidden Markov Model</i>	70
Figura 7.5 – Fluxograma do <i>Hidden Markov Model</i>	71
Figura 7.6 – Topologia em linha	73
Figura 7.7 – Topologia com duas rotas possíveis	73
Figura 7.8 – Topologia com um nó intermediário para encaminhamento	74
Figura 7.9 – <i>Handoff</i> topologia 1 com AODV	76
Figura 7.10 – <i>Handoff</i> topologia 2 com AODV	77
Figura 7.11 – <i>Handoff</i> topologia 3 com AODV	77
Figura 7.12 – Latência topologia 1 com AODV	78
Figura 7.13 – Latência topologia 1 com OLSR	79
Figura 7.14 – Latência topologia 2 com AODV	79
Figura 7.15 – Latência topologia 2 com OLSR	79
Figura 7.16 – Latência topologia 3 com AODV	80
Figura 7.17 – Latência topologia 3 com OLSR	80
Figura 7.18 – Taxa de entrega topologia 1 com AODV	81
Figura 7.19 – Taxa de entrega topologia 1 com OLSR	81

Figura 7.20 – Taxa de entrega topologia 2 com AODV	82
Figura 7.21 – Taxa de entrega topologia 2 com OLSR	82
Figura 7.22 – Taxa de entrega topologia 3 com AODV	83
Figura 7.23 – Taxa de entrega topologia 3 com OLSR	83

LISTA DE TABELAS

Tabela 5.1 – Modelos que utilizam o processo de Markov	50
Tabela 6.1 – Formato de mensagens OLSR	63
Tabela 7.1 – Exemplo de uma amostra dos dados brutos coletados	65
Tabela 7.2 – Exemplo de uma amostra dos dados pré-processados	66
Tabela 7.3 – Classificação da frequência em relação ao ruído base	67
Tabela 7.4 – Definição das classes para a classificação dos dados das amostras	67
Tabela 7.5 – Exemplo de parâmetros de um Modelo	71
Tabela 7.6 – Configuração da RNA Marques (2016)	72
Tabela 7.7 – Padrões de geração de tráfego	74
Tabela 7.8 – Definição das classes para a métrica de <i>Handoff</i>	76

LISTA DE SIGLAS

ACK	<i>Acknowledgement</i>
ANATEL	Agência Nacional de Telecomunicações
BC	Base do Conhecimento
CCC	Canal de Controle Comum
DV	Descoberta de vizinhos
FUTEBOL	<i>Federated Union of Telecommunications Research Facilities for EU-Brazil Open Laboratory</i>
HMM	<i>Hidden Markov Model</i>
IA	Inteligência Artificial
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
ISM	<i>Industrial, Scientific and Medical</i>
MAC	<i>Medium Access Control</i>
MDE	Modelo de Decisão do Espectro
M	<i>Master</i>
ME	Mobilidade do Espectro
MPL	<i>Multilayer Perceptron</i>
QoS	<i>Quality of Services</i>
RC	Rádio Cognitivo
RNA	Rede Neural Artificial
RnF	<i>Random Forest</i>
RRC	Rede de Rádios Cognitivos

RREQ	<i>Route Requests</i>
RREP	<i>Route Reply</i>
RSSF	Rede de Sensores Sem Fio
RSSI	<i>Received Signal Strength Indication</i>
S	<i>Slave</i>
SDN	<i>Software Defined Network</i>
SDR	<i>Software Defined Radio</i>
XML	<i>Extensible Markup Language</i>
WINET	<i>Wireless Networking</i>

SUMÁRIO

1	Introdução	11
1.1	Definição do Problema	13
1.2	Solução Proposta	14
1.3	Contribuições	14
1.4	Estrutura do Trabalho	15
2	Referencial Teórico	16
2.1	Coexistência	16
2.2	Rádios Cognitivos	16
2.2.1	Sensoriamento de Espectro	20
2.2.2	Decisão do Espectro	22
2.2.3	Compartilhamento do Espectro	23
2.2.4	Mobilidade do Espectro	26
2.3	Aprendizado de Máquina	26
2.4	Rádio Definido por Software	30
2.5	Plataformas de Desenvolvimento	31
2.5.1	GNU <i>Radio</i>	31
2.6	Ambiente de Desenvolvimento e Testes	32
3	Trabalhos Relacionados	34
3.1	Arquitetura SIRC<i>o</i>	37
4	<i>Framework</i> para Rádios Cognitivos	40
4.1	Arquitetura	40
4.2	Operação do <i>Framework</i>	42
5	Métodos de Decisão baseados em aprendizado de máquina	44
5.1	<i>Random Forest</i>	44
5.1.1	Árvore de decisão	44
5.1.2	Definição do <i>Random Forest</i>	47
5.2	<i>Hidden Markov Model</i>	47
5.2.1	Definição do Modelo HMM	51
5.2.2	Baum Welch	52
5.2.3	Algoritmo de Viterbi	53
5.3	Rede Neural Artificial	55

6	Protocolos de Roteamento	58
6.1	<i>Ad Hoc On-Demand Distance Vector</i>	58
6.2	<i>Optimized Link State Routing Protocol</i>	62
7	Metodologia de desenvolvimento e Resultados	65
7.1	Modelagem dos métodos de aprendizado de máquina	65
7.1.1	<i>Random Forest</i>	66
7.1.2	<i>Hidden Markov Model</i>	70
7.1.3	Rede Neural Artificial	72
7.2	Topologias	73
7.3	Métricas avaliadas	75
7.4	Resultados	75
7.4.1	<i>Handoff</i>	75
7.4.2	Latência	78
7.4.3	Taxa de Entrega	81
8	Conclusão	85
	REFERÊNCIAS	87

1 INTRODUÇÃO

Nos últimos anos, foi observado um crescimento acelerado no uso dos dispositivos sem fio. Esse aumento é devido a popularidade dos dispositivos IoT, aplicações como automação residencial, redes de sensores sem fio, telefonia celular e inúmeras outras aplicações que, em sua maioria, utilizam as faixas ISM (*Industrial, Scientific and Medical*). Devido ao baixo custo e ampla utilização de dispositivos de comunicação sem fio, o espectro disponível tornou-se insuficiente para suportar as necessidades deste grande número de dispositivos. Esse aumento de usuários leva à coexistência de um grande número de dispositivos em uma mesma faixa de frequência, o que pode ocasionar interferências prejudiciais ao desempenho da rede.

As faixas de frequência são definidas por Drozd et al. (2005) como um hiperespaço teórico ocupado por sinais de rádio, que possuem dimensões de localização, frequência, tempo, entre outras possibilidades. Esse hiperespaço é chamado de eletro-espço, hiperespaço de transmissão, espaço de espectro de rádio ou simplesmente espaço de espectro, e pode ser utilizado para descrever como o ambiente de rádio é dividido entre diversos sistemas (HORNE, 2003).

As faixas ISM são faixas de frequência de livre utilização e os dispositivos que as utilizam são denominados Usuários Secundários. Em contrapartida, faixas licenciadas são as faixas de frequência que exigem uma licença para sua utilização e os dispositivos que as utilizam são denominados Usuários Primários. No Brasil, a ANATEL (Agência Nacional de Telecomunicações) regulamenta que faixas ISM devem empregar as faixas de frequência de 900 MHz; 2,4 GHz; 5,8 GHz, sendo que a 2,4 GHz é amplamente utilizada pelo Wi-Fi assim como o Bluetooth. Da mesma maneira, a 5,8 GHz tem se tornado mais utilizada em redes domésticas, uma vez que a nova tecnologia do Wi-Fi abrange essa faixa.

De acordo com Zhou, Stankovic e Son (2006), o nível de ocupação do espectro está sujeito a sofrer problemas de sobreposição, dado o crescimento da quantidade de dispositivos de comunicação sem fio. Esse aumento na quantidade de dispositivos sem fio deve-se principalmente ao avanço tecnológico e na popularidade de dispositivos para a *Internet of Things* (IoT). Tais equipamentos são utilizados para uma infinidade de aplicações, como em redes de sensores sem fio, automação industrial, RFID e redes domésticas. Em maio de 2016 a empresa Google lançou o *Google Home*, dispositivo onde o usuário tem a possibilidade de controlar todos os equipamentos sem fio em sua rede doméstica por comandos de voz (STONEHEM, 2016). Esse tipo de tecnologia mostra como se encontra o avanço tecnológico hoje, além da maneira como

dispositivos sem fio estão sendo amplamente utilizados e fazendo o uso das faixas ISM para a transferência de dados.

Estas limitações motivaram uma mudança de paradigma de utilização estática do espectro para um gerenciamento dinâmico do espectro. O objetivo é detectar uma determinada faixa do espectro que não esteja em utilização, e em seguida usá-la para transmissão evitando sofrer interferência. Para que isso seja possível, os equipamentos utilizam dispositivos capazes de realizar o acesso dinâmico do espectro (*Dynamic Spectrum Access - DSA*), onde os rádios monitoram continuamente o espectro e são reconfigurados para acessar de maneira oportunista as faixas não utilizadas ou subutilizadas (AKYILDIZ et al., 2006).

O desenvolvimento da tecnologia das telecomunicações permitiu a transição de rádios baseados apenas em *hardware* para rádios que utilizam uma combinação de *software* e *hardware*. Mitola (1992) definiu esses rádios como *Software Defined Radios* (SDR) que são rádios formados por um circuito de rádio frequência que utiliza *software* para a modulação e demodulação do sinal. Com isso, foi possível acoplar o SDR com algum tipo de inteligência artificial possibilitando a criação de Rádios inteligentes ou Cognitivos.

Os Rádios Cognitivos são rádios capazes de realizar de maneira autônoma a reconfiguração da rede por meio de aprendizado e adaptação do meio em que se encontra inserido (MITOLA; MAGUIRE, 1999). Os Rádios Cognitivos possuem a capacidade de, alterando as configurações do rádio, realizar transmissões em faixas de espectros licenciadas, sem que interfiram na transmissão dos Usuários Primários, assim como em faixas não licenciadas, que não estejam em grande utilização. Redes com Rádios Cognitivos são capazes de prover alta largura de banda devido às suas características de sensoriamento e análise do espectro (AKYILDIZ et al., 2008). Segundo o FCC (2003), existe um grande interesse no desenvolvimento de dispositivos de Rádio Cognitivo, pois essa tecnologia tem potencial de aumentar a eficiência do uso do espectro em um momento que a demanda por comunicações sem fio está em um crescimento acelerado.

Devido ao avanço em pesquisas de Rádios Cognitivos, diversos *frameworks* para o desenvolvimento de Redes de Rádios Cognitivos (RRC) têm sido propostos. Os *frameworks*, em sua maioria, foram desenvolvidos e avaliados em ambientes simulados. Esses ambientes foram desenvolvidos para observar o comportamento de redes sem fio convencionais, de modo que as características dinâmicas do espectro não são corretamente modeladas no contexto de Rádios Cognitivos (RIZVI; SHOWANI, 2014). Além disso, alguns *frameworks* focam no desenvolvi-

mento apenas de parte das atividades de uma Rede de Rádios Cognitivos, como por exemplo no sensoriamento do espectro. Outro ponto a ser levado em consideração é o fato dos *frameworks* realizarem implementação apenas nas camadas física e de enlace, o que não possibilita que exista a comunicação *multi-hop* na Rede de Rádios Cognitivos.

1.1 Definição do Problema

Dado a grande utilização das faixas ISM, fica evidente o problema da sobreposição e interferência de transmissões nesses canais de comunicação livre. A coexistência de diferentes redes que utilizam frequências próximas leva a interferências prejudiciais, ou até mesmo a impedir completamente o funcionamento da rede (FLAMMINI et al., 2009). O uso da frequência de 2,4 GHz em alguns locais já atinge quase o limite máximo do canal, e estudos realizados pela Shared Spectrum Company (2010) mostram que existe uma necessidade em identificar faixas adicionais de espectro para aplicações de banda larga móvel. Esse problema da superutilização de faixas específicas do espectro ao mesmo tempo em que outras faixas são subutilizadas, ocorre principalmente pelo fato de poucas redes oferecerem suporte necessário para a utilização de Rádios Cognitivos.

Além disso, o uso de aprendizado de máquina, que consiste em técnicas de manipulação de dados com o objetivo de identificar padrões, tem se tornado cada vez mais comum em trabalhos referentes ao sensoriamento e decisão de espectro. Entretanto, esses trabalhos em sua maioria são desenvolvidos e avaliados através de simulações por causa de sua complexidade. Tais simulações muitas vezes utilizam bases de dados artificiais, o que muitas vezes diverge de um cenário real, de modo que um modelo de um cenário simulado não seria capaz de representar corretamente a realidade. Outro fator é a escolha de métodos pelos *frameworks* para definir a faixa de espectro a ser utilizada pela RRC. Muitas vezes esses *frameworks* utilizam métodos simples baseados somente na força do sinal recebido *received signal strength indicator* (RSSI), ou algum algoritmo de Inteligência Artificial (IA) menos robusto, ou ainda um algoritmo robusto porém não adequadamente treinado para reconhecer os padrões da rede. Isso pode gerar resultados indesejados, onde o método utilizado indicaria faixas do espectro que não são as melhores em determinada situação.

Os *frameworks* conhecidos para o desenvolvimento de Redes de Rádios Cognitivos não possuem implementações que contemplem a comunicação multi-hop da RRC. De modo geral, existem diversos trabalhos em que modelos analíticos são propostos para a implementação de

um protocolo de roteamento, porém em nenhum deles são realizadas simulações, e tão pouco testes reais, para validar os modelos propostos.

1.2 Solução Proposta

Este trabalho propõe a extensão da arquitetura desenvolvida por (MARQUES et al., 2016), uma arquitetura para o sensoriamento inteligente de Rádios Cognitivos. O trabalho proposto visa o desenvolvimento de um *framework* para criação de Redes de Rádios Cognitivos que implemente todas as etapas de execução de um Rádio Cognitivo. O *framework* é capaz de corretamente executar o sensoriamento, decisão, compartilhamento e mobilidade do espectro.

Esse *framework* contempla ainda a criação de dois novos métodos para decisão do espectro baseado em aprendizado de máquina. Os métodos de decisão do espectro desenvolvidos são baseados em *Random Forest* e *Hidden Markov Model*. Esses métodos escolhem uma faixa de frequência do espectro em determinado momento que seja a mais adequada para as transmissões da RRC, de maneira que evite ao máximo interferências com outras redes.

O *framework* ainda oferece suporte a comunicação multi-hop da RRC. Esse suporte é realizado através da implementação de dois protocolos desenvolvidos para a camada de rede. O primeiro protocolo utilizado é o *Ad hoc On-Demand Distance Vector* (AODV) um protocolo reativo que cria rotas entre nós de destino apenas se eles forem solicitados pelos nós de origem. O segundo protocolo utilizado é o *Optimized Link State Routing* (OLSR) um protocolo pró-ativo baseado em estados de enlace que limita a quantidade de nós capazes de encaminhar pacotes na rede, o que elimina mensagens redundantes.

1.3 Contribuições

Como contribuição direta relacionada a este trabalho, foi publicado no *15th International Symposium on Wireless Communication Systems - ISWCS*, o trabalho intitulado *Analysis of machine learning algorithms for spectrum decision in cognitive radios* (MARTINS PINTO; CORREIA, 2018).

Como contribuição indireta relacionada a este trabalho, foi apresentado no *XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, o minicurso intitulado *Experimental Wireless Networking Research using Software-Defined Radios* (SOUZA et al., 2017).

1.4 Estrutura do Trabalho

O Capítulo 2 apresenta um referencial teórico com os principais conceitos abordados neste trabalho. Esse capítulo relata o problema de coexistência, explica a solução dos Rádios Cognitivos e suas etapas de sensoriamento, decisão, compartilhamento e mobilidade do espectro. Descreve algoritmos de aprendizado de máquina, assim como os algoritmos de roteamento. Além disso, fala sobre Rádios Definidos por Software, sobre plataformas de desenvolvimento e um exemplo de ambiente de desenvolvimento. O Capítulo 3 fala sobre diversos *frameworks* de desenvolvimento de RRC e explica a arquitetura original de onde esse trabalho foi baseado. O Capítulo 4 apresenta o *framework* proposto explicando as mudanças desenvolvidas. O Capítulo 5 explica o desenvolvimento dos métodos de decisão do espectro baseados em algoritmos de aprendizado de máquina. O Capítulo 6 apresenta os modelos de implementação dos protocolos de roteamento implementados. O Capítulo 7 retrata a metodologia de desenvolvimento e discute sobre os resultados encontrados. O Capítulo 8 apresenta as conclusões obtidas com este trabalho e as propostas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado o referencial teórico onde são abordados os conceitos utilizados para a elaboração deste trabalho. Primeiramente é abordado o problema da coexistência em redes sem fio e como os Rádios Cognitivos podem auxiliar na redução desse problema. Os principais conceitos de Rádios Cognitivos são explicados, além de alguns métodos de aprendizado que máquina que podem ser utilizados na etapa de Decisão do Espectro.

2.1 Coexistência

O espectro utilizado para comunicações sem fio é um recurso limitado. Desse modo, é necessário que ele seja utilizado de forma eficiente de modo a atender o maior número possível de usuários. A solução deste problema é um dos grandes desafios da engenharia de telecomunicações.

O limitado espectro de frequências poderá ainda ser completamente congestionado devido ao desenvolvimento de diversos dispositivos de comunicação sem fio de baixa potência. Esses dispositivos têm se tornado populares em diversas áreas, desde controle de sistemas industriais até mesmo automações residenciais. Devido ao uso e implantação de maneira imprevisível desses tipos de redes, aliado a equipamentos elétricos que também interferem no espectro como por exemplo o forno de micro-ondas, a rede será extremamente congestionada (ZHOU; STANKOVIC; SON, 2006).

O trabalho feito por McHenry et al. (2006) realizou medições em Chicago, Illinois - uma das áreas mais densamente povoadas nos Estados Unidos. As Medições foram feitas entre as faixas de 30 MHz a 3000 MHz. Com os resultados, os autores verificaram que a ocupação do espectro variou de 1% da capacidade máxima do canal até mais de 70% da capacidade máxima do canal. Puderam verificar ainda que durante o período das medições mais de 80% do espectro estava sendo desperdiçado.

A variação de utilização do canal em conjunto com partes do espectro que não foram utilizadas são informações de grande relevância para o desenvolvimento dos Rádios Cognitivos.

2.2 Rádios Cognitivos

De acordo com a norma FCC (2003), um Rádio Cognitivo é um rádio que tem a capacidade de alterar seus parâmetros de transmissão baseados na interação com o ambiente de

operação, podendo ajustar seus parâmetros operacionais de maneira autônoma, de forma a maximizar o desempenho da rede, diminuir a interferência e facilitar a interoperabilidade.

O termo Rádio Cognitivo foi definido por Mitola e Maguire (1999) como tecnologias de rádio que tornam possível o uso mais intensivo e eficiente do espectro por usuários licenciados em suas próprias redes e inclusive por usuários que compartilham o acesso ao espectro, em uma base negociada ou oportunista. Dessa maneira, as redes que fazem uso de Rádio Cognitivo possuem como principal objetivo a maximização do uso do espectro e redução da interferência, provendo uma alta taxa de transmissão em uma rede heterogênea que acessa dinamicamente o espectro (PRINCIPE; CHALASANI, 2014; MITOLA; MAGUIRE, 1999).

A tecnologia de Rádio Cognitivo possui a habilidade de oferecer benefícios que resultarão em um maior número de acessos ao espectro, assim como melhorar os serviços de comunicação disponíveis. Essa tecnologia, possui ainda a capacidade de dividir o espectro, cooperando eficientemente com outros usuários. Akyildiz, Lee e Chowdhury (2009) denominam partes do espectro que não estejam sendo utilizadas como "buracos no espectro" ou "espaços em branco". O Rádio Cognitivo identifica essas partes do espectro e transmite nesses espaços livres, resultando assim em um uso mais eficiente do espectro enquanto evita interferências provenientes de outros usuários.

A habilidade do Rádio Cognitivo em adaptar o uso do espectro de acordo com as condições em tempo real do ambiente, oferece às entidades reguladoras, aos usuários licenciados e aos usuários em geral, uma maior flexibilidade e eficiência no uso do espectro disponível. Por esse motivo, reduz-se o risco de interferências que prejudiquem a rede (FCC, 2003).

Como descrito na norma FCC (2003) a tecnologia de Rádio Cognitivo pode ser utilizada para melhorar a eficiência do uso do espectro em diversos cenários. O primeiro exemplo citado ocorre quando um usuário licenciado utiliza o Rádio Cognitivo em sua própria faixa de espectro para melhorar o desempenho de sua rede. Uma outra situação ocorre quando o Rádio Cognitivo é utilizado para possibilitar que usuários não licenciados possam acessar uma faixa licenciada de espectro, de maneira que o usuário não licenciado entrará em operação somente em momentos ou localidades onde o espectro licenciado não se encontra em uso pelo usuário licenciado.

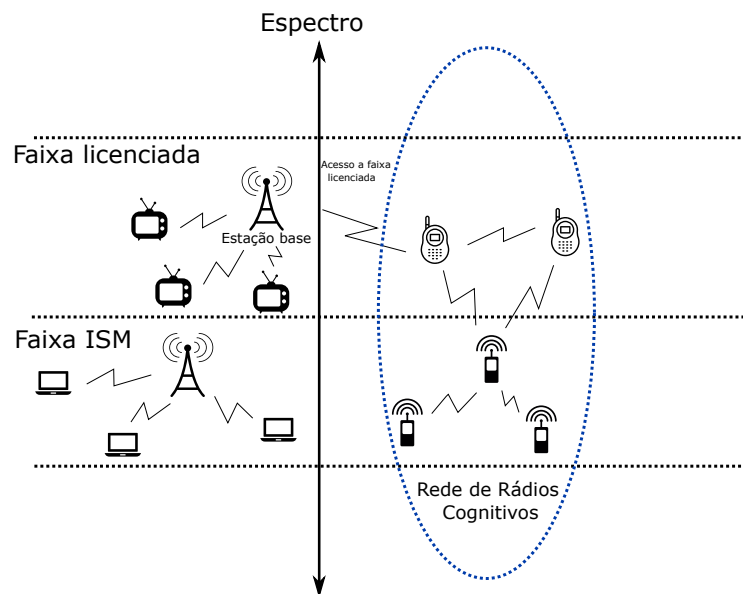
Os benefícios do uso de Rádio Cognitivo está na habilidade do mesmo em permitir o uso de alta transmissão por usuários não licenciados, em áreas rurais ou áreas onde o espectro seja pouco utilizado, e assim facilitar a interoperabilidade entre diferentes sistemas de transmissão

sem fio, oferecendo, a partir disto, uma maior implantação de dispositivos heterogêneos em um mesmo ambiente.

Dispositivos de comunicação sem fio são divididos em dois grupos distintos de usuários, são eles os Usuários Primários e Usuários Secundários. Os Usuários Primários são aqueles que possuem a licença de utilização da faixa do espectro, podendo realizar transmissões sem qualquer tipo de restrição. Por outro lado, os Usuários Secundários, não possuem tal licença, e por isso não podem realizar transmissões quando Usuários Primários estiverem utilizando sua faixa de espectro. Dessa forma, sempre que um Usuário Secundário desejar realizar uma transmissão em uma faixa de espectro licenciada, ele deverá primeiramente realizar o sensoriamento do ambiente, a fim de verificar a existência de atividades de Usuários Primários. Com isso, evita-se que ocorram interferências nas transmissões, possibilitando uma comunicação confiável.

A Figura 2.1 exemplifica a interação entre Usuários Primários e Secundários em uma Rede de Rádios Cognitivos. Os usuários de Rádios Cognitivos interagem com os Usuários primários de uma rede de diversas maneiras.

Figura 2.1 – Arquitetura de uma rede de rádios cognitivos baseado em Akyildiz et al. (2008)



1. **Acesso à rede de Rádio Cognitivos** - os usuários de Rádios Cognitivos podem acessar a estação base tanto na faixa licenciada, quanto na faixa não licenciada. Todas as interações ocorrem dentro da rede de Rádios Cognitivos, que utiliza métodos para evitar o uso de uma faixa licenciada enquanto o Usuário Primário dessa faixa estiver utilizando a mesma.

2. **Acesso Ad Hoc** - usuários de Rádios Cognitivos são capazes de comunicar com outros usuários de Rádios Cognitivos através de uma conexão Ad Hoc, tanto nas faixas licenciadas quanto nas não licenciadas.
3. **Acesso à rede primária** - usuários de Rádios Cognitivos podem também utilizar a estação base do Usuário Primário em uma faixa licenciada. Contudo, essa conexão é realizada utilizando um protocolo de controle de acesso ao meio adaptativo, de forma que seja realizado o *roaming* a fim de evitar o uso da mesma faixa de espectro do Usuário Primário.

Para que o Rádio Cognitivo seja capaz de realizar tal comunicação, é necessário que quatro funcionalidades sejam empregadas para o gerenciamento do espectro. São elas: *sensoriamento do espectro*, *decisão do espectro*, *compartilhamento do espectro* e *mobilidade do espectro* (DUDLEY et al., 2014).

Akyildiz, Lee e Chowdhury (2009) explicam cada uma dessas funcionalidades da seguinte maneira:

- **Sensoriamento de Espectro** (*Spectrum Sensing*): O Rádio Cognitivo deve monitorar as faixas de espectro, guardar essas informações e então detectar os espaços em branco, ou seja, as faixas que não estão sendo utilizadas. Essa é uma funcionalidade básica do Rádio Cognitivo e totalmente ligada ao gerenciamento do espectro, oferecendo assim, informações sobre a disponibilidade do espectro.
- **Decisão de Espectro** (*Spectrum Decision*): Após identificar faixas de espectro disponíveis, o Rádio Cognitivo deve selecionar a melhor faixa disponível de acordo com os critérios determinados que variam de acordo com a aplicação, a fim de realizar a escolha da faixa a ser utilizada.
- **Compartilhamento de Espectro** (*Spectrum Sharing*): As transmissões do Rádio Cognitivo devem ser coordenadas pelo compartilhamento de espectro para evitar colisões entre os usuários da rede. A alocação de canal e de potência são realizadas a fim de evitar que sejam causadas interferências aos Usuários Primários.
- **Mobilidade de Espectro** (*Spectrum Mobility*): Em uma faixa licenciada de espectro, o usuário de Rádio Cognitivo deve trocar para outra faixa de espectro quando um Usuário Primário exigir o uso do canal. Esse processo é chamado de *Spectrum Handoff*. Já

em canais não licenciados, esse processo será necessário quando for identificado uma diminuição do desempenho da rede, que pode ser causada por grande número de usuário transmitindo em uma determinada região do espectro. Dessa maneira, é necessário realizar o *Spectrum Handoff*, ou mudança para outra faixa mais adequada.

Para a utilização dessas funcionalidades de gerenciamento de espectro, uma possível abordagem é a utilização de um canal de controle comum, de forma a possibilitar que todos os dispositivos possam trocar informações de controle.

2.2.1 Sensoriamento de Espectro

A proposta do sensoriamento de espectro está na aquisição de dados do ambiente, com o objetivo de analisar o estado que se encontram um ou mais canais em um determinado período de tempo. O sensoriamento de espectro é uma parte crucial para a tecnologia de Rádio Cognitivo, para que, de maneira eficiente e precisa, seja possível detectar a presença de Usuários Primários e evitar que exista interferência com os mesmos (YU et al., 2012). A eficiência do sensoriamento do espectro está diretamente relacionada à utilização do espectro em Rádios Cognitivos (CHEN; QIU, 2010). Ainda de acordo com Yu et al. (2012), o desempenho do sensoriamento de espectro é caracterizado pela precisão e a eficiência do sensoriamento.

- A precisão do sensoriamento se refere a exatidão na detecção de sinais de Usuários Primários, de maneira que suas transmissões não sofram interferências. Essa precisão é determinada pela probabilidade da ocorrência de um alarme falso e a probabilidade de detecção do Usuário Primário.
- Eficiência do sensoriamento se refere à quantidade de faixas de espectro disponíveis que são encontradas utilizando uma certa quantidade de recursos computacionais, ou seja, quanto mais faixas encontradas de acordo com uma certa quantidade de *overhead* de sensoriamento, maior será a eficiência do sensoriamento.

Essas duas características de precisão do sensoriamento e eficiência do sensoriamento, são aspectos opostos que refletem o desempenho do sensoriamento de espectro. Uma maior precisão do sensoriamento exigirá que a rede seja sensoreada por mais tempo, isso resultaria em um *overhead* maior que reflete na diminuição da eficiência do sensoriamento. Como o desempenho geral do sensoriamento de espectro depende da precisão e eficiência do sensoriamento,

é necessário que haja um balanceamento entre essas duas características. O nível de complexidade para a realização do sensoriamento do espectro varia de acordo com o método que é utilizado, influenciando diretamente na precisão e eficiência do sensoriamento. A obtenção de alguns parâmetros usados no sensoriamento, como modulação ou amplitude do sinal, exige uma maior complexidade computacional (YUCEK; ARSLAM, 2009). Portanto, as técnicas devem ser cuidadosamente analisadas para que as suas utilizações não se tornem inviáveis.

O sensoriamento de espectro foi alvo de diversas pesquisas nos últimos anos, devido aos diversos desafios que existem sobre esse assunto. Um exemplo, como citados em Zeng et al. (2010) são as variações existentes nos níveis de interferência e ruído dependendo do tempo e da localização. Por esse motivo, alguns métodos de sensoriamento de espectro foram sugeridos na literatura.

Existem diferentes métodos para o sensoriamento do espectro, e eles podem ser divididos em Sensoriamento Local e Sensoriamento Cooperativo. No Sensoriamento Local, um único usuário faz o sensoriamento de forma independente. Por outro lado, no modelo cooperativo, o sensoriamento individual de cada usuário é compartilhado com diversos outros usuários, para que a precisão do sensoriamento seja maior, aumentando assim a confiabilidade (AKYILDIZ et al., 2008). O Sensoriamento Cooperativo é realizado como uma alternativa ao Sensoriamento Local, para o aumento da precisão do sensoriamento, por meio do compartilhamento de informações sobre o espectro entre diversos Usuários Secundários presentes na rede. Um problema clássico de redes sem fio, o problema do terminal escondido, é eliminado através desse tipo de Sensoriamento Cooperativo. Como explicado em Marinho e Monteiro (2012) existem trocas de informações entre os diversos Usuários Secundários, de forma que mesmo que um determinado Usuário Secundário não esteja ao alcance do sinal de um Usuário Primário, ele receberá informações sobre a atividade dos Usuários Primários por meio da cooperação entre Usuários Secundários. Por meio dessa cooperação, é possível ainda definir a região de alcance dos Usuários Primários, o que resolve o problema do terminal escondido.

Atualmente, a principal dificuldade na utilização do sensoriamento de espectro cooperativo está na definição da quantidade de informações que deverão ser enviadas pela rede, quando essas informações devem ser trocadas e quais os usuários da rede deverão sensoriar o espectro. Esse compartilhamento de informação aumenta a quantidade de transmissões pela rede, o que pode atrapalhar o funcionamento normal da rede (PATIL; PATIL, 2016).

De acordo com Min Song et al. (2012), Cabric, Mishra e Brodersen (2004), Dudley et al. (2014) os algoritmos de sensoriamento de espectro são divididos em três esquemas: detecção de energia, detecção de filtro combinado e detecção de características.

- Na detecção de energia é realizada uma amostragem do valor da intensidade do sinal no espectro (*RSSI - Received Signal Strength Indicator*), e se esse valor for superior a um valor pré-definido, então existirão transmissões sendo realizadas nesse espectro.
- Na detecção de filtro combinado, característica do Usuário Primário como modulação, frequência e formato dos pacotes de dados são previamente conhecidas. Assim, o Usuário Secundário ao identificar essas características na rede, já reconhece a presença do Usuário Primário, reduzindo assim o tempo de sensoriamento.
- Na detecção de características, o método utilizado parte do princípio que os sinais seguem um padrão periódico, de forma que os parâmetros estatísticos, como a média, variam de forma periódica. É realizada uma análise de uma função de correlação espectral, e essa correlação é bem definida nos sinais dos Usuários Primários, ao contrário do sinal de ruído que não possui essa correlação espectral (ÖNER; JONDRAL, 2007). Dessa forma, existe uma distinção clara entre o sinal e ruído.

2.2.2 Decisão do Espectro

Uma vez que são encontradas faixas de frequências disponíveis no espectro durante a etapa de sensoriamento, o próximo passo é decidir se existe ou não a necessidade de realizar a troca de canal da rede. As redes de Rádio Cognitivo utilizam faixa do espectro, de maneira que transmissões realizadas não prejudiquem os Usuários Primários (AKYILDIZ et al., 2008).

Similarmente à fase de sensoriamento de espectro, na etapa de decisão do espectro existem duas abordagens, são elas: abordagem distribuída, onde as decisões são tomadas de forma independente pelos usuários presentes na rede; ou uma abordagem centralizada, onde as informações sobre o sensoriamento são enviadas para um coordenador que informará a decisão tomada aos outros usuários presentes na rede.

Caso seja necessário realizar a mudança de faixa, a definição de qual faixa de espectro escolher é realizada levando em consideração certas características, são elas:

- **Interferência:** A quantidade de interferência no canal verificada pelo receptor.

- **Perda de caminho:** A perda de caminho trata da relação entre a frequência e a distância. Se a frequência aumenta, há uma maior perda do caminho, ou seja, o alcance da transmissão é menor. No caso em que haja o aumento da potência para compensar a perda do caminho, isso pode ocasionar interferências em outros usuários.
- **Erros na camada de enlace:** A taxa de erro do canal varia de acordo com o esquema de modulação e o nível da interferência presente no espectro.
- **Atraso na camada de enlace:** Diversos tipos de protocolos são utilizados na camada de enlace, para lidar com as diferentes situações de perda de caminho, erros na camada de enlace e interferência (SAIFAN; KAMAL; GUAN, 2012). Tudo isso resulta em diferentes atrasos da camada de enlace.

Descentralizar a decisão de alocação do espectro aumenta a escalabilidade e auxilia na redução da complexidade, pois com o aumento do número de dispositivos na rede, a mesma se torna computacionalmente muito complexa para um único nó central realizar a decisão de alocação do espectro (RENK et al., 2008; MARINHO; MONTEIRO, 2012).

A etapa do Sensoriamento do Espectro possui grande importância, pois nessa etapa é realizada a coleta de informações do espectro que será utilizada na etapa de decisão. A etapa de Decisão do Espectro é onde está a inteligência do rádio, uma vez que são utilizadas técnicas de aprendizado de máquina para a escolha do melhor canal.

2.2.3 Compartilhamento do Espectro

A natureza da rede sem fio exige uma coordenação nas tentativas de transmissão entre os usuários de Rádios Cognitivos. Por causa disso, o compartilhamento do espectro deve utilizar diversas funcionalidades do protocolo de controle de acesso ao meio - *Medium Access Control* (MAC) (ZHANG; YU, 2010; PANDIT; SINGH, 2013). Além disso, a característica singular de coexistência de usuários de Rádios Cognitivos com Usuários Primários cria diferentes desafios para o compartilhamento de espectro em Redes de Rádios Cognitivos.

A primeira classificação do Compartilhamento do Espectro é baseada na arquitetura que pode ser centralizada ou distribuída.

- **Compartilhamento do espectro centralizado:** Existe uma entidade central a qual controla os procedimentos de Compartilhamento do Espectro. Contudo, um procedimento de

sensoriamento do espectro distribuído pode ser utilizado, desde que as informações sejam encaminhadas para uma entidade central a qual deverá construir o mapa do espectro (ILERI et al., 2005).

- **Compartilhamento do espectro distribuído:** Alocação e acesso do espectro são realizadas localmente por cada nó de forma distribuída (ZHAO et al., 2007).

A segunda classificação do compartilhamento do espectro é baseada no comportamento de alocação do espectro, que é classificado em cooperativo e não cooperativo.

- **Compartilhamento cooperativo do espectro:** O compartilhamento cooperativo utiliza as medidas de interferência de cada nó de maneira que os efeitos da comunicação entre os nós são considerados. A técnica comum consiste na formação de *clusters* que compartilham as informações de interferência localmente, e isso oferece um equilíbrio entre esquemas centralizados e distribuídos.
- **Compartilhamento não cooperativo do espectro:** Apenas um nó é considerado no compartilhamento não cooperativo do espectro (ZHENG; CAO, 2005). Por não considerar as interferências de outros nós, esse tipo de compartilhamento não cooperativo está sujeito a utilização reduzida do espectro. Contudo, esse tipo de solução não exige uma frequente troca de mensagens entre os nós vizinhos, como é feito nas soluções cooperativas.

Abordagens cooperativas geralmente são superiores em desempenho das abordagens não cooperativas e tendem a resultados próximos ao melhor resultado global (PENG; ZHENG; ZHAO, 2006). Além disso, as técnicas cooperativas também possuem um maior grau de justiça, além de maior taxa de transferência. Por outro lado, o desempenho menor das técnicas não cooperativas são compensados pela menor quantidade de informação trocada e conseqüentemente menor consumo de energia.

A terceira classificação do compartilhamento do espectro é baseada nas técnicas de acesso ao espectro que é dividido em sobreposição e subjacência.

- **Sobreposição do compartilhamento do espectro:** Os dispositivos de Rádio Cognitivo ouvem as mensagens enviadas pelo Usuário Primário e usam essas mensagens para reduzir a interferência. Os nós acessam a rede utilizando partes do espectro que não foram utilizadas pelo Usuário Primário, desse modo, minimizando interferências com a rede primária.

- **Subjacência do compartilhamento do espectro:** São permitidas transmissões simultâneas de usuários de Rádio Cognitivo e usuários comuns, desde que o nível de interferência no lado do Usuário Primário permaneça aceitável. As técnicas de espalhamento do espectro são utilizadas de modo que a transmissão de um nó do Rádio Cognitivo é considerado ruído pelo Usuário Primário.

Técnicas de subjacência podem utilizar maior largura de banda ao custo de maior complexidade da rede. Além disso, técnicas híbridas podem ser consideradas para melhorar o desempenho de técnicas de acesso ao espectro (MENON; BUEHRER; REED, 2005).

A quarta classificação do compartilhamento do espectro é baseada no escopo e dividida em intra ou inter-rede.

- **Compartilhamento de espectro intra rede:** É o compartilhamento dentro de uma rede de Rádios Cognitivos. Essa abordagem foca na alocação entre as entidades da rede de Rádios Cognitivos. Desse modo, usuários da rede de Rádios Cognitivos acessam o espectro sem causar interferência aos Usuários Primários.
- **Compartilhamento de espectro inter-rede:** É o compartilhamento entre diversas redes de Rádios Cognitivos. A arquitetura da rede de Rádios Cognitivos possibilita que diversas redes consigam coexistir em um mesmo ambiente. Desse modo, o compartilhamento inter-rede possibilita o compartilhamento entre redes de Rádios Cognitivos distintas, incluindo políticas que sejam necessárias.

Existem diversos desafios de pesquisas para a realização de uma operação eficiente e transparente do compartilhamento do espectro. Dentre os mais populares estão o canal de controle comum, distância da faixa de rádio dinâmica e informação de localização.

- **Canal de controle comum (CCC):** O canal de controle comum auxilia diversas funcionalidades do compartilhamento do espectro. Contudo, nas redes de Rádios Cognitivos, um canal de controle comum a todos os usuários é altamente dependente da topologia e varia ao longo do tempo, e por isso deve levar em consideração um canal que esteja sempre disponível (ZHAO; ZHENG; YANG, 2005). Por causa disso, técnicas para reduzir a utilização de um canal de controle comum devem ser criadas, ou então, um canal de controle comum pode ser explorado localmente para um conjunto específico de nós.

- **Distância da faixa de rádio dinâmica:** Devido a interdependência entre a distância da faixa de rádio e a frequência de operação, os vizinhos de um nó podem mudar uma vez que a frequência de operação se altera.
- **Informação de localização:** Assumimos que em redes de Rádios Cognitivos o Usuário Secundário sempre sabe a localização e a potência de transmissões do Usuário Primário, dessa maneira, cálculos de interferência podem ser realizados de forma mais fácil. Contudo, essa presunção não é considerada sempre válida em redes de Rádios Cognitivos. (AKYILDIZ et al., 2008).

2.2.4 Mobilidade do Espectro

Após o Rádio Cognitivo encontrar o melhor espectro disponível, a atividade do Usuário Primário nesse determinado espectro pode exigir que o usuário de Rádio Cognitivo altere sua faixa de operação, essa atividade é denominada mobilidade do espectro. Dessa maneira, o Usuário Primário pode exigir que o Usuário Secundário altere sua faixa de transmissão para outra porção livre do espectro (CACCIAPUOTI; AKYILDIZ; PAURA, 2013).

O propósito da mobilidade de espectro em redes de Rádios Cognitivos é assegurar a mudança de faixas do espectro de forma suave e rápida, de maneira a diminuir ao máximo a degradação do desempenho durante essa troca de faixas (CHEN; HONG, 2013). Porém, esse *handoff* do espectro gera atrasos na transmissão, levando a perdas de pacotes e degradação da taxa de transmissão no espectro (SONG; XIE, 2012).

2.3 Aprendizado de Máquina

Por causa do espectro ser um meio compartilhado, as transmissões de um dispositivo podem ser identificadas por todos os outros em uma mesma frequência de transmissão. Se um dispositivo detectar duas ou mais transmissões ao mesmo tempo, esse dispositivo não consegue distinguir as transmissões e obter as informações desejadas. Essa é uma situação de interferência, que é um dos focos dos métodos de decisão do espectro. Os métodos de decisão do espectro devem determinar frequências onde exista baixa interferência. Como mencionado anteriormente, os Rádios Cognitivos podem utilizar métodos de aprendizado de máquina para realizar a etapa de decisão do espectro.

Aprendizado de máquina de acordo com Griesmer e McCarthy (1992), é o estudo de sistemas que aprendem por meio de dados. O aprendizado de máquina pode ser utilizado para aprender por experiências, dado um conjunto de tarefas e medidas de desempenho, a fim de aumentar o desempenho em um ambiente desejado.

Como explicado por Katidiotis, Tsagkaris e Demestichas (2010), existem três abordagens distintas para aprendizado de máquina, são elas: Aprendizado Supervisionado, Aprendizado Não Supervisionado e Aprendizado Semi-Supervisionado.

- O Aprendizado Supervisionado procura criar regras de decisão baseadas em um conjunto de amostras conhecidas e bem definidas. Esse conjunto de amostras é denominado Base de Treinamento. Essas amostras, que também podem ser chamados de Instâncias, são compostas por um ou mais valores de entrada e um valor de saída. Esses valores são representados por um valor numérico, que determinam a qual classe essa instância faz parte. Esses valores são representados por um intervalo no conjunto dos números inteiros ou reais.
- O Aprendizado Não Supervisionado, diferentemente do Aprendizado Supervisionado, não possui uma base de dados, de maneira que as regras de decisão são desenvolvidas durante sua operação. Essa técnica de aprendizado de máquina utiliza dados brutos ou pré-processados em conjuntos de valores de entrada, porém não possuem um valor de saída como é utilizado no Aprendizado Supervisionado. O foco desse tipo de aprendizado de máquina está na busca de dados similares, para que seja possível definir uma classe para a criação das regras de decisão.
- O Aprendizado Semi-Supervisionado utiliza a técnica de ambos aprendizados vistos anteriormente, de forma que exista uma base de treinamento a ser utilizada antes da ativação do sistema. Essa base pode ser modificada com inclusão de novos dados em tempo de execução.

Katidiotis, Tsagkaris e Demestichas (2010) ainda explicam que existem dois tipos principais de algoritmos que são os algoritmos de Classificação e Algoritmos de Regressão.

- Os Algoritmos de Classificação possuem um valor de saída bem definido por um conjunto de classes determinado, que são criadas pelo usuário de acordo com o tipo de problema que deseja ser solucionado. Essas classes não possuem um limite em relação à

quantidade, contudo, estão diretamente relacionadas com o problema a ser solucionado. Dependendo da modelagem do problema abordado, esse número de classe pode ser limitado em relação aos recursos computacionais disponíveis.

- Os Algoritmos de Regressão possuem um valor de saída definido por um valor numérico em um intervalo do conjunto dos números reais. Nesse caso, o principal fator para a limitação desse tipo de algoritmos está na precisão dos valores que se deseja obter. Quanto mais a precisão for necessária, maior será o uso de recursos computacionais para a resolução do problema. Dependendo da precisão necessária para resolução de um problema específico, o consumo dos recursos computacionais podem influenciar diretamente no tempo de processamento dos dados e conseqüentemente da execução do Algoritmos de Regressão.

A principal vantagem na utilização de algoritmos de aprendizado de máquina em Rádio Cognitivo é a capacidade de modelar o comportamento da rede de modo a prever os intervalos futuros, permitindo dessa maneira que soluções possam ser aplicadas em tempo hábil. Um bom exemplo de uma aplicação nesse contexto é a troca de canal (BANTOUNA et al., 2012). Uma outra alternativa é proposta por Zhao e Swami (2007) para aplicações onde as informações do meio não podem ser obtidas em tempo real. Nesse caso, é então definido um modelo estático simplificado de uso do espectro por outros usuários do mesmo meio.

Um lado negativo ao utilizar aprendizado de máquina está na potencial demanda por uma grande quantidade de recursos computacionais requeridos, seja pela estrutura de dados a ser utilizada, ou pela forma como os dados são manipulados, de maneira a obter a previsão desejada. Existem alguns algoritmos de aprendizado de máquina que necessitam utilizar toda a base de dados de treinamento durante a sua execução, o que exige que a mesma seja armazenada, consumindo assim os recursos computacionais. Existem ainda, casos onde o tratamento de dados mais complexos sejam necessários, o que faz com que a obtenção da previsão seja demorada ou até mesmo inviável.

Em relação ao aprendizado de máquina no contexto de Rádio Cognitivo, o objetivo é a otimização de métricas da Qualidade de Serviço (*Quality of Service - QoS*). Bantouna et al. (2012), explica que uma métrica bastante utilizada é o *bitrate*, que informa a quantidade de dados transmitidos ao longo de um espaço de tempo. Um valor maior do *bitrate* indica que houve uma maior quantidade de dados transferidos, o que indica um bom resultado.

Jayaweera et al. (2011) mostraram o conceito de ODAL (*Observe, Decide, Act, Learn* - Observar, Decidir, Agir, Aprender) como sendo o Rádio Cognitivo do futuro. Nesse trabalho, foi mostrado que os rádios serão capazes de automaticamente aprender com os erros, e dessa maneira, convergir para uma configuração ideal. Os rádios ainda consideram uma série de parâmetros como, por exemplo, atraso de comunicação, precisão no sensoriamento, consumo de energia, segurança, entre outros. Essa experiência adquirida pelo Rádio Cognitivo na parte de decisão, possibilita que as ações futuras sejam cada vez mais otimizadas, dessa maneira, esse objetivo pode ser alcançado utilizando o aprendizado de máquina.

Métodos Cooperativos

Em aprendizado de máquina, um conceito amplamente aceito é que o desempenho de um conjunto de classificadores simples é geralmente melhor que um único classificador, dada a mesma quantidade de dados de treinamento (SIRIKULVIRIYA; SINTHUPINYO, 2011). Os métodos mais populares que utilizam conjuntos de dados são conhecidos por *boosting* (FREUND; SCHAPIRE, 1996), *bagging* (BREIMAN, 1996) e *Random Forest*.

O método *boosting* cria diferentes bases de aprendizado, alterando os pesos de forma sequencial das instâncias no conjunto de treinamento. No início, todas as instâncias são inicializadas com pesos idênticos. Cada instância classificada incorretamente pela base de treinamento receberá um peso maior na próxima etapa de classificação, de modo a tentar classificá-lo corretamente. O erro é então calculado, o peso das instâncias classificadas corretamente são diminuídos e o peso das instâncias calculadas incorretamente são aumentados. Assim, o voto de cada um dos classificadores possui peso proporcional ao seu desempenho (WANG et al., 2011).

O método *bagging* (*bootstrap aggregation*), possui diferentes subconjuntos de treinamento que são aleatoriamente selecionados para substituir o conjunto de treinamento. Cada subconjunto de treinamento é utilizado como base de treinamento para uma instância do método. Assim, todas as instâncias treinadas com o subconjunto de base de treinamento são combinadas e o resultado final é a maior quantidade de votos obtidos por cada uma das instâncias (BREIMAN, 1996). O *boosting* consegue gerar os classificadores sequencialmente, enquanto que o *bagging* gera os classificadores em paralelo.

O método *Random Forest* é um outro método que utiliza subconjuntos, que consiste em construir diversas árvores de decisão que serão utilizadas para classificar uma nova instância, utilizando a maioria dos votos. Cada nó das árvores de decisão utiliza um subconjunto de

atributos aleatoriamente selecionados de todo o conjunto original de atributos. Além disso, cada árvore utiliza uma amostra de dados diferente, da mesma maneira que o método *bagging* (BREIMAN, 2001).

De forma geral, o método *bagging* é na maioria das vezes mais preciso que um único classificador, porém às vezes tem um desempenho menor que o método *boosting*. Por outro lado, o método *boosting* consegue criar subconjunto que são menos precisos que um único classificador. Em certas situações, o método *boosting* leva em consideração no treinamento os ruídos presentes na amostra de dados e por isso o desempenho é prejudicado. O método Random Forest, por outro lado, é mais robusto que o *boosting* em relação ao ruído, mais rápido que o *bagging* e o *boosting* e seu desempenho é tão bom quanto ou até mesmo melhor que o método *boosting*, além de não sofrer com *overfitting* (BREIMAN, 2001)

O erro de uma *Random Forest* depende da força individual de cada árvore na floresta, e a correlação entre duas árvores na floresta; a força pode ser entendida como uma medida de desempenho de cada árvore. Dessa forma, aumentar a correlação aumenta a taxa de erro da floresta, e aumentar a força das árvores individuais diminui a taxa de erro da floresta, na medida em que uma árvore com uma baixa taxa de erro é um classificador forte (YAQUB et al., 2011).

A quantidade de árvores em uma *Random Forest* reflete na quantidade de recursos computacionais utilizados para o método. Em alguns casos, aumentar a quantidade de árvores em uma *Random Forest* apenas aumenta a quantidade de recursos computacionais utilizados, com praticamente nenhuma melhora no desempenho do método. A quantidade ideal gira em torno de 64 a 128 árvores dependendo do problema em questão, de maneira que não existe ganho significativo de desempenho a partir dessa quantidade, além de existir um equilíbrio entre o desempenho, tempo de processamento e uso de memória (OSHIRO; PEREZ; BARANAUSKAS, 2012).

2.4 Rádio Definido por Software

Ao longo dos últimos anos, cada vez mais atenção tem sido dada à incorporação de novas capacidades de processamento computacional em tecnologia de sistemas de rádio. Os sistemas de rádio estão incorporando cada vez mais *softwares* em sua arquitetura, com isso, esses sistemas possuem a habilidade de adaptar seu comportamento, baseados em fatores diversos do ambiente em que se encontram.

Comumente, os rádios comercializados possuem um microprocessador e um *software* para controlar parâmetros operacionais, como frequência e modulação, porém esse *software* não poderia ser alterado. Os desenvolvedores de rádios estão incorporando capacidades para a programação de *softwares* nos rádios, e isso possibilita que funções básicas possam ser alteradas com maior facilidade (DILLINGER; MADANI; ALONISTIOTI, 2003). De acordo com FCC (2003) *software defined radio* (SDR), ou Rádio Definido por *Software*, é um dispositivo onde os parâmetros de operação são controlados por *software*, permitindo assim que o rádio seja programado para transmitir e receber em diversas frequências, ou até mesmo utilizar mais de um formato de transmissão, desde que seja suportado pelo *hardware*.

2.5 Plataformas de Desenvolvimento

Para que o Rádio Cognitivo seja programado, são necessárias as alterações de diversos parâmetros, contudo essa programação pode se tornar custosa. Por esse motivo, a utilização de ferramentas que auxiliem no desenvolvimento dos *softwares* para Rádio Cognitivo se faz necessária. Uma plataforma bem conhecida é o GNU Radio.

2.5.1 GNU Radio

O GNU *Radio* é uma plataforma de *software* livre com código aberto, que permite o desenvolvimento dos componentes de processamento de sinais por meio de blocos de processamentos, que são necessários para a implementação de Rádios Definidos por *Software*, e podem utilizar *hardware* de baixo custo, ou em ambientes de simulação. O GNU *Radio* possui a licença de utilização GNU *General Public License* (GPL), e seus direitos autorais pertencem à Fundação de *Software* Livre (*Free Software Foundation* - FSF) (BLOSSOM, 2016).

As aplicações do GNU *Radio* são desenvolvidas utilizando a linguagem de programação Python, enquanto o processamento de sinais, por ser mais crítico em relação ao desempenho, é implementado utilizando a linguagem de programação C++, por ser uma linguagem de programação com um melhor desempenho. Com isto, é possível realizar a implementação de sistemas de rádio de alto desempenho que fazem uso da eficiência da linguagem C++, juntamente com a simplicidade do desenvolvimento de aplicações em Python (BLOSSOM, 2016). O GNU *Radio* ainda oferece suporte para o desenvolvimento de algoritmos de processamento de sinais, utilizando dados previamente armazenados ou gerados por meio de *Software*, o que permite que seja utilizado sem *hardware* caso haja necessidade.

O GNU *Radio* oferece ao desenvolvedor uma biblioteca de blocos de processamento do sinal e uma estrutura para realizar a conexão entre os blocos. O desenvolvimento é realizado por meio de uma estrutura de grafos, onde os vértices representam os blocos de processamento de sinais e as arestas representam o fluxo de dados entre os blocos de processamento (BLOSSOM, 2016). De modo geral, os blocos de processamentos de sinais são desenvolvidos na linguagem C++, enquanto a estrutura de grafos é desenvolvida utilizando a linguagem Python.

Diversos tipos de *hardware* são compatíveis com o GNU *Radio*. Grande parte dos equipamentos que possuam um conversor analógico digital podem ser utilizados. Existe a compatibilidade com a família de dispositivos USRP (*Universal Software Radio Peripheral*) que é desenvolvida pela *Ettus Research* especificamente para ser utilizado com o GNU *Radio*, apesar de existirem adaptações para mais plataformas, como o *Simulink* e *LabVIEW*.

A *Ettus Research* é uma empresa líder em fornecimento de plataformas para SDR, sendo a USRP seu principal produto. As USRP são plataformas de SDR de baixo custo que são equipadas com interfaces que chegam a frequências de até 6GHz e possuem aplicações diversas como: telefonia celular ou até mesmo transmissão de televisão (ETTUS, 2017).

Neste projeto foi utilizada a versão do GNU *Radio* mais recente, assim como a USRP para testes em ambientes reais.

Possui e manter um ambiente onde exista todos os recursos disponíveis para esse tipo de trabalho no desenvolvimento de Rádios Cognitivos pode ser custoso, uma vez que essa é uma tecnologia relativamente recente e equipamentos de hardware podem custar muito caro. Assim, caso a aquisição desses recursos não seja viável um ambiente de desenvolvimento e teste pode ser utilizado remotamente.

2.6 Ambiente de Desenvolvimento e Testes

Projeto FUTEBOL (*Federal Union of Telecommunications Research Facilities for an EU-Brazil Open Laboratory*), é um projeto de pesquisa com cooperação internacional entre Brasil e União Europeia, na forma de um consórcio com as instituições desses países. O projeto recebe financiamento da iniciativa *European Union's Horizon 2020* para pesquisa, desenvolvimento tecnológico e do Ministério de Ciência Tecnologia e Inovação, através da RNP e CTIC¹. Esse projeto tem como meta desenvolver e implantar uma infra-estrutura de pesquisa asso-

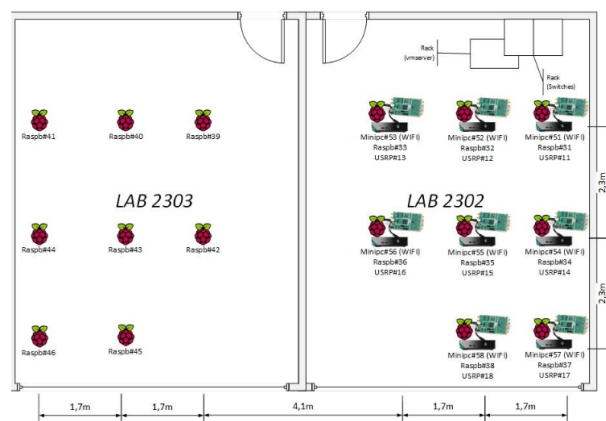
¹ <http://www.ict-futebol.org.br/>

ciado a um *framework* para experimentações na Europa e no Brasil, que possibilite pesquisas experimentais com a utilização de redes ópticas e redes sem fio.

Esse projeto possui uma *testbet* no departamento de Ciência da Computação da Universidade Federal de Minas Gerais (UFMG) que oferece um laboratório para a realização de pesquisas em Rede Sem Fio e *Internet of things* (IoT). Esse laboratório possibilita que os equipamentos disponibilizados sejam utilizados sem custos pelos pesquisadores que possuem cadastro no *Fed4Fire*². Dessa maneira, as instituições de pesquisa de todo o mundo podem utilizar remotamente equipamentos de ponta, de modo que qualquer experimento possa ser configurado e disparado de qualquer lugar do mundo. Por isso, as pesquisas em redes de celulares, redes Wi-Fi, redes *bluetooth* e IoT poderão ser feitas sem a necessidade dos pesquisadores estarem fisicamente próximos aos equipamentos necessários.

O FUTEBOL é um ambiente para experimentos com tecnologia sem fio, Redes Definidas por Software (*Software Defined Network* - SDN), Rádios definidos por Software (*Software Defined Radio* - SDR), Internet das coisas (IoT) e integração de redes ópticas e redes sem fio. O ambiente de desenvolvimento conta com duas salas que pertencem ao laboratório WINET³ (salas 2302 e 2303). A disposição dos equipamentos disponíveis pode ser encontrada pela Figura 2.2.

Figura 2.2 – Mapa dos equipamentos implantados DCC/UFMG (FUTEBOL, 2018)



² <https://www.fed4fire.eu/>

³ <http://www.winet.dcc.ufmg.br>

3 TRABALHOS RELACIONADOS

Nos últimos anos, diversos *frameworks* para Redes de Rádios Cognitivos foram idealizados e desenvolvidos com o uso de simuladores como por exemplo o NS, OMNet ++, QUALNET e outros (SUNDMAN, 2008). Contudo, grande parte desses simuladores foram projetados com a finalidade de observar o comportamento das redes sem fio tradicionais, não sendo capazes de modelar e simular corretamente as características dinâmicas do espectro para os Rádios Cognitivos (RIZVI; SHOWANI, 2014).

Rizvi e Showani (2014) apresentaram um modelo analítico para a construção de um *framework* para simulação de uma rede de Rádios Cognitivos baseada em camadas. Um simulador baseado em eventos discretos foi considerado para a implementação do *framework*. Apesar das discussões da modelagem, nenhum resultado completo da simulação de uma rede de Rádios Cognitivos foi apresentado.

O trabalho de Kim et al. (2007), processa e classifica os sinais em uma determinada frequência com o objetivo de identificar os padrões do espectro. Para classificação, foi utilizado *Hidden Markov Model* (HMM) para processar os sinais identificados devido à sua robusta capacidade de detecção de padrões. Apesar dos autores afirmarem que foram capazes de detectar e classificar sinais em uma faixa de espectro mesmo com a relação de *signal-to-noise* baixos, não foi apresentada nenhuma aplicação nas Redes de Rádios Cognitivos e os testes foram realizados por meio de simulações.

Os autores López-Benítez e Casadevall (2014) propuseram um *framework* para modelagem e simulação multidimensional da ocupação do espectro. Uma modelagem unificada permite que as propriedades de ocupação do espectro nas dimensões de tempo, frequência e espaço sejam simuladas simultaneamente. O *framework* permite a avaliação das técnicas de acesso dinâmico ao canal para as redes de Rádios Cognitivos. Contudo, os modelos idealizados e o *framework* não foram considerados para implementação em plataformas reais de Rádios Cognitivos.

Os autores Jararweh et al. (2014) propuseram um *framework* que utiliza o conceito de hipervisores de multicamadas para alocação eficiente de recursos. Esse *framework* introduz um esquema de controle semi-decentralizado que permite que a estação base da Rede de Rádios Cognitivos delegue algumas das responsabilidades de gerenciamento aos usuários da rede. É ainda realizada a virtualização de recursos da Rede de Rádios Cognitivos, o que permite a alocação de recursos dinâmicos, livres de infraestrutura e eficientes para os usuários. Entre-

tanto, o desenvolvimento e a avaliação de desempenho do *framework* foi realizado por meio de simulações.

Um *framework* para redes de sensores sem fio que empregam Rádios Cognitivos foi proposto em Correia et al. (2012). O objetivo foi o desenvolvimento e teste de protocolos de decisão do espectro para seleção dinâmica do melhor canal, considerando os requisitos da aplicação. Resultados no simulador Castalia mostraram que os protocolos de decisão do espectro melhoraram a taxa de entrega, mantendo a latência e o consumo de energia. Este *framework* não considerou a implementação dos Rádios Cognitivos em hardware real.

Um *framework* desenvolvido para SDR usando GNU *Radio* foi proposto por Jagannath, Saarinen e Drozd (2015). O objetivo era a implementação de módulos para desenvolvimento e testes de novas técnicas de classificação automática de múltiplos sinais. Apesar da precisão na classificação de sinais, e da importância no sensoriamento do espectro, os autores não apresentaram sua aplicação nas Redes de Rádios Cognitivos.

Um *survey* foi realizado por Bkassiny, Li e Jayaweera (2013). Neste trabalho, foram caracterizados os problemas de aprendizagem em Rádios Cognitivos e foi apresentado a importância da inteligência artificial na comunicação cognitiva de sistemas reais. Este trabalho analisa diversos problemas de aprendizagem que foram estudados no contexto de RC e estes foram classificados em duas categorias principais, sendo elas a tomada de decisão e a classificação de recursos. A tomada de decisão é responsável por determinar políticas e regras de decisão para os RC, enquanto a classificação de características permite identificar e classificar diferentes modelos de observação. Os algoritmos de aprendizado encontrados são categorizados como algoritmos supervisionados ou não supervisionados. O trabalho descreve em detalhes vários desafios de aprendizado que surgem em Redes de Rádio Cognitivo, e são apresentados possíveis métodos para resolvê-los. Por fim, é discutida as semelhanças e diferenças entre os algoritmos apresentados e as condições sob as quais cada uma das técnicas pode ser aplicada.

Christodoulou (2009) mostrou que um dispositivo que utiliza Radio Cognitivo pode ser treinado para reconfigurar automaticamente suas antenas e então se adaptar ao ambiente em que se encontra inserido, além de explicar os conceitos de ODAL (Observe, Decide, Act, Learn). Esse trabalho, entretanto, não apresentou o uso de um método de aprendizado de máquina ou alguma implementação em uma Rede de Rádios Cognitivos.

O trabalho realizado por Lu et al. (2016) utiliza K-means e SVM para criar um vetor de probabilidade de baixa dimensão para classificação baseada em aprendizado de máquina. Esse

método proposto converte um vetor de característica com muitas dimensões em um vetor de característica bidimensional constante para técnicas de aprendizado de máquina, mantendo o mesmo desempenho de detecção de espectro. Devido à sua menor dimensão, a classificação baseada em vetor de probabilidade é capaz de ter menor duração de treinamento e menor tempo de classificação para vetores de teste. Este trabalho utilizou simulações além do foco ser somente na etapa de sensoriamento do espectro.

No trabalho de Huang et al. (2009) foi projetado um mecanismo de aprendizado baseado em support vector machine (SVM). A abordagem proposta é demonstrada por meio de simulações realizadas no MATLAB e pelos dados provenientes da plataforma do protocolo 802.11a. Os resultados de classificação e regressão do SVM foram promissores, garantindo a eficácia do mecanismo de aprendizagem. Contudo, o trabalho além de utilizar ambientes simulados ainda considerou para validação dos testes apenas a camada física, além de não demonstrar sua utilização em um Rede de Rádios Cognitivos.

No trabalho de Saad, Staehle e Knorr (2016) é apresentado um algoritmo de predição do espectro baseado em HMM. Este trabalho cita abordagens tradicionais que utilizam HMM com apenas dois estados ocultos, o que permite a previsão limitada de apenas um passo ao futuro. Dessa maneira, este trabalho mostra que estender a quantidade de estados ocultos e modelar o problema como uma classificação de *maximum likelihood* permite um intervalo de predição de múltiplos intervalos no futuro. Contudo, este trabalho apenas apresenta análises da etapa de sensoriamento, além de realizar apenas simulações para validação dos resultados.

O trabalho desenvolvido por Sharma e Bohara (2014) discutiu vários algoritmos de aprendizado de máquina e suas aplicações na resolução de problemas específicos de rádio cognitivo. Os algoritmos analisados foram os Algoritmos Genéticos (AG), Redes Neurais Artificiais (RNA) e Hidden Markov Model (HMM). Foi afirmado que os modelos que utilizam RNA e HMM são mais adequados para problemas de previsão e classificação de RC onde os dados de treinamento são necessários para resolver o problema, além de concluir que o HMM não sofre *overfitting* e ainda que esse modelo é melhor para quantidades de amostras pequenas. Contudo, esse trabalho não realizou implementações em RRC para validar as análises realizadas.

No trabalho de Marques (2016) foi apresentada a arquitetura SIRCo. Essa arquitetura foi desenvolvida para Redes de Rádios Cognitivos que possui um módulo de decisão do espectro, onde foram utilizados métodos para escolha do melhor canal do espectro. Nesse módulo foi utilizada uma Rede Neural Artificial. Essa RNA foi treinada considerando uma base de dados

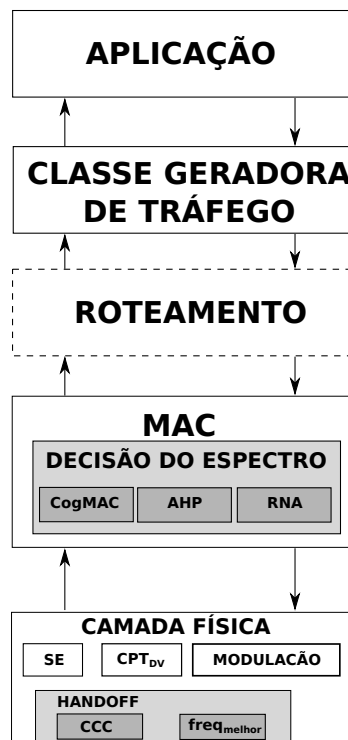
local, garantindo assim uma maior largura de banda. A arquitetura foi avaliada em ambiente real e os dispositivos utilizados foram as *Universal Software Radio Peripheral* (USRP), modelos B200 e B210. Contudo, essa arquitetura implementada utilizou poucas abordagens diferentes para os métodos de aprendizado de máquina. Além disso, a comunicação é feita somente no nível da camada MAC.

O trabalho proposto baseia-se na extensão dessa arquitetura, visando preencher as lacunas existentes com a ampliação do módulo de decisão do espectro, utilizando HMM e RnF, além do suporte a comunicação *multi-hop* com AODV e OLSR. Apesar de (MARQUES, 2016) afirmar que a técnica de HMM é computacionalmente complexa e por esse motivo não é aconselhável para uma RRC em um ambiente dinâmico e real, a etapa de treinamento de um HMM equipara-se à técnica da RNA, onde o treinamento é realizado com uma base de treinamento antes de ser utilizado na RRC. Desse modo, somente o modelo previamente treinado é utilizado na RRC.

3.1 Arquitetura SIRCo

A arquitetura proposta por Marques et al. (2016) foi dividida em camadas e pode ser visualizada pela Figura 3.1.

Figura 3.1 – Arquitetura do protocolo SIRCo Marques (2016).

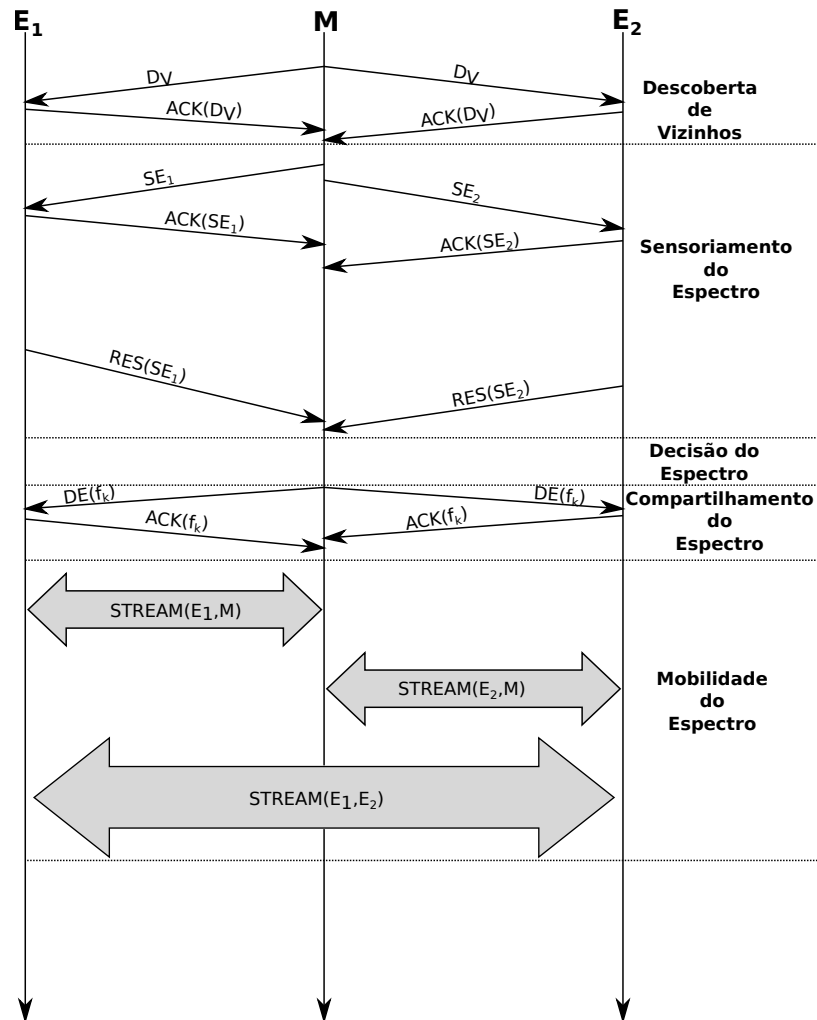


A camada mais superior, camada de aplicação, determina as mensagens geradas pela arquitetura. Estas mensagens são informações de tamanho fixo e são compostas por conjuntos aleatórios de *bits*. Em seguida, a arquitetura conta com uma classe geradora de tráfego que gera padrões de tráfego a serem utilizados durante a transmissão de dados. Na camada de rede não houve a implementação de protocolos de roteamento, de fato, não há implementação alguma da camada de rede. A camada MAC é composta pelos métodos de decisão do espectro. Por fim, na camada física é encontrado a implementação do sensoriamento do espectro, modulação, um método de controle de potência para descoberta de vizinhos e um módulo que realiza a reconfiguração dos rádios.

Essa arquitetura utiliza um modelo baseado no paradigma *Master e Slave*. O nó *Master* é o responsável pelo gerenciamento da Rede de Rádios Cognitivos, pois é nele que se encontra os métodos de decisão, além de ser o responsável pela requisição do sensoriamento da rede e compartilhamento das informações de melhor canal. Os nós *Slaves* realizam o sensoriamento do espectro e enviam essa informação para o nó *Master*.

O SIRCo inicia seu ciclo de execução com uma requisição de descoberta de vizinhos feita pelo nó *Master* que é respondida pelos nós *Slaves*. Após receber a confirmação, o nó *Master* solicita a realização do sensoriamento do espectro, que é realizado pelos nós *Slaves*. O nó *Master* recebendo as informações referente ao sensoriamento do espectro inicia a etapa de decisão do espectro, onde os métodos de decisão devem escolher a melhor frequência disponível. Com a melhor frequência disponível determinada pelo método de decisão, o nó *Master* compartilha essa informação com os nós *Slaves* que então migram para a frequência informada, caso seja necessário. Finalmente, após todos os nós estarem na mesma faixa de frequência, a etapa de transmissão de dados é iniciada e sua duração é de sessenta segundos. Após esse intervalo o ciclo reinicia. A Figura 3.2 ilustra o ciclo de execução com o nó *Master* sendo representado por *M* e os nós *Slaves* sendo representados por *E*₁ e *E*₂.

Figura 3.2 – Fluxo de mensagens SIRCo Marques (2016).



O Sensoriamento Inteligente para Rádios Cognitivos (SIRCo) é uma arquitetura para RRC que utiliza métodos cognitivos para a coexistência de diversas redes heterogêneas em um mesmo ambiente. Essa arquitetura utiliza um método para a decisão do espectro baseado em Rede Neural Artificial (RNA). O framework desenvolvido neste trabalho, que é uma extensão desse trabalho onde é acrescentado novos métodos de decisão baseados em aprendizado de máquina além de suporte a comunicação multi-hop é apresentado no próximo capítulo.

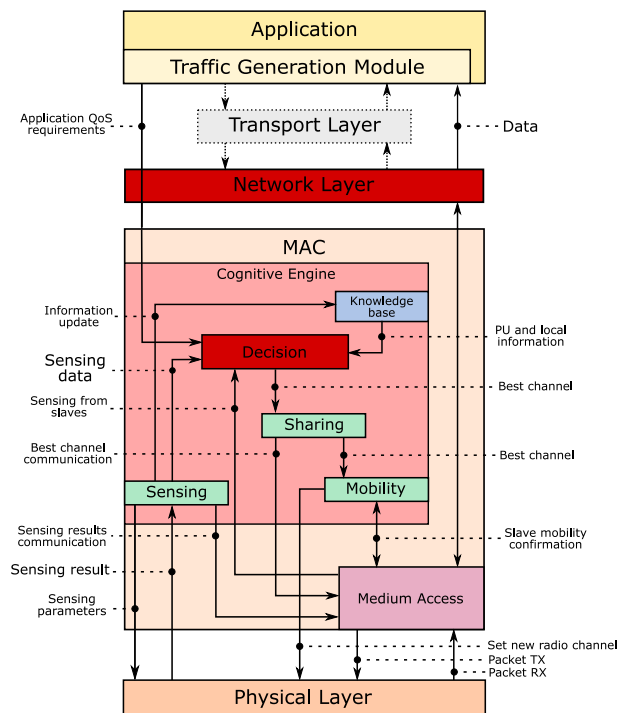
4 FRAMEWORK PARA RÁDIOS COGNITIVOS

Este capítulo apresenta um *framework* que expande a arquitetura apresentada por Marques et al. (2016). Este *framework* conta com dois novos métodos de decisão do espectro, baseados em aprendizado de máquina HMM e RnF, além do suporte a comunicação multi-hop com a implementação do AODV e OLSR.

4.1 Arquitetura

Como mencionado anteriormente, a arquitetura proposta por Marques et al. (2016) foi expandida. Foram adicionados dois novos métodos de decisão, bem como implementações de protocolos de roteamento na camada de rede. Na Figura 4.1 são mostradas as principais camadas da arquitetura, que são baseadas em um modelo de camada *cross-layer*. Os blocos destacados em vermelho representam os principais contribuições implementadas nesse trabalho. As informações dessas camadas são descritas a seguir.

Figura 4.1 – Arquitetura do *framework* adaptado de Souza et al. (2017)



- **Aplicação** - Define os requisitos de qualidade de serviço, os tipos de mensagens e os tipos de tráfego suportados pelo *framework*. As políticas de qualidade de serviço são estabelecidas pela aplicação para determinar os requisitos relacionados à largura de banda,

latência e taxa de entrega. Essas políticas influenciam no comportamento da camada MAC em relação à frequência de detecção, ciclo operacional e parâmetros do rádio.

- **Módulo Gerador de Tráfego** - Gera padrões de tráfego usados para troca de mensagens. Este módulo foi desenvolvido para avaliação do modelo, gerando tráfego que simula o comportamento de uma aplicação real. Três distribuições diferentes foram implementadas para o intervalo de envio de pacotes: exponencial, uniforme e constante. Além disso, este módulo inclui distribuições contínuas e discretas.
- **Transporte** - Não foi implementado. Como os experimentos se concentraram em sensoriamento, decisão, compartilhamento e mobilidade do espectro em um cenário de comunicação multi-hop, os serviços fornecidos por essa camada puderam ser momentaneamente desconsiderados. No entanto, o *framework* permite a inserção de módulos adicionais que podem incluir recursos da camada de transporte.
- **Rede** - Permite a comunicação multi-hop da rede e a execução do roteamento dos pacotes entre a origem e o destino. Dois protocolos foram utilizados para a implementação dessa camada, AODV - (*Ad Hoc On-Demand Distance Vector*) e OLSR (*Optimized Link State Routing Protocol*).
- **MAC** - Responsável por evitar colisões e controlar a transmissão de pacotes originados das camadas superiores. Além disso, esta camada implementa os métodos de decisão do espectro baseados em Rede Neural Artificial (RNA), *Random Forest* (RnF) e *Hidden Markov Model* (HMM).
- **Física** - Todas as informações sobre a intensidade do sinal, localização e parâmetros suportados pelo rádio são coletadas por esta camada. Além disso, possui funções para ajustar os parâmetros do rádio e transmitir e receber pacotes por meio de um canal sem fio. A etapa de sensoriamento do espectro é realizado nesta camada dentro de uma faixa de 800 MHz a 5,8 GHz.

Na camada MAC está presente o motor cognitivo que consiste nos quatro estágios de um Rádio Cognitivo e uma base de conhecimento. A base de conhecimento consiste nas informações coletadas por meio do sensoriamento do espectro local. As informações da base de conhecimento podem ser atualizadas dinamicamente por dados coletados da detecção de espec-

tro local. O *framework* usará essas informações como entrada para os métodos de decisão do espectro.

O principal bloco do motor cognitivo é a Decisão de Espectro (DE), onde todos os métodos de aprendizado de máquina são implementados. Este *framework* fornece três métodos para a decisão do espectro: *Hidden Markov Model* (HMM), Rede Neural Artificial (RNA) e *Random Forest* (RnF). Esses métodos de decisão são afetados pelos parâmetros de entrada: QoS, sensoriamento do espectro e a base de conhecimento. Com base nessas entradas, os métodos decidem qual canal faz o melhor uso do espectro. Após a DE escolher o melhor canal, é necessário realizar o compartilhamento dessas informações com os outros nós presentes na rede. Essa etapa é realizada pelo módulo de compartilhamento do espectro.

O módulo de compartilhamento do espectro comunica o canal definido como o melhor pela DE para os demais nós. Neste contexto, um canal de controle comum de 6 GHz (CCC) é usado para inicializar a comunicação entre nós e trocar mensagens de controle como um canal de retorno. A escolha deste canal é definida na fase de decisão do espectro. A troca de mensagens também é controlada pelo módulo de acesso ao meio.

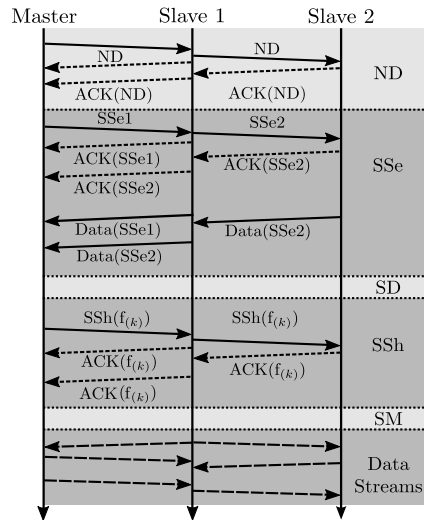
Informações sobre o melhor canal também são usadas no bloco mobilidade do espectro (ME). O *Master* informa o melhor canal para todos os nós *Slaves* e depois de receber pelo menos uma confirmação de um nó *Slave*, o nó *Master* migra para o canal informado.

4.2 Operação do *Framework*

Neste *framework*, o modelo de comunicação utiliza o paradigma *Master-Slave*. Para implementar as quatro fases da RRC utilizou-se a arquitetura apresentada anteriormente pela Figura 4.1. A Figura 4.2 ilustra um exemplo de um ciclo de operação do *framework*, utilizando um nó *Master* (M) e dois nós *Slaves* (S_1 e S_2). Note que o nó S_1 encaminha os pacotes entre o nó M e o nó S_2 .

O fluxo de mensagens é iniciado pelo nó M, que envia mensagens em *broadcast* para iniciar a fase de descoberta do vizinho *neighbor detection* (ND). O nó S_1 recebe a requisição de ND e reencaminha essa requisição para o nó S_2 assim como para qualquer outro nó que estiver ao seu alcance. Após o nó S_1 reencaminhar a requisição para seus vizinhos, é gerado uma resposta que é encaminhada para o nó *Master*. Quando o nó S_2 envia sua resposta de ND o nó S_1 simplesmente encaminha essa resposta para o nó M. Ao final dessa etapa, todos os nós da rede conhecem seus vizinhos de primeiro salto.

Figura 4.2 – Fluxo de Mensagens entre 3 nós



Em seguida, o nó M envia mensagens solicitando o sensoriamento do espectro (SSe_n) para todos os nós. Essa mensagem é respondida pelos nós *Slaves* com um $ACK(SSe_n)$. Após receber a resposta $ACK(SSe_n)$, o nó M fica aguardando os dados referentes ao resultado do sensoriamento.

Ao final do sensoriamento realizado pelos nós S_1 e S_2 , as informações do espectro são enviadas para o nó M para análise do método de decisão do espectro. Nesse momento é realizada a etapa de *Spectrum Decision* (SD). Durante a fase SD, a melhor frequência é escolhida pelo método de decisão selecionado para esse fim. Esse método pode ser a RNA, o RnF ou o HMM.

Na fase de Compartilhamento de Espectro *Spectrum Sharing* (SSh), o M envia a informação do melhor canal escolhido para seus vizinhos, que é respondida pelos nós *Slaves* com $ACK(f_{(k)})$. Neste momento os nós da rede possuem a informação do melhor canal disponível, desta maneira, os nós podem então migrar para a nova frequência, finalizando assim essa fase. A comunicação é por fim iniciada e pode ser realizada bidirecionalmente tanto de *Master* para *Slaves*, quanto entre nós *Slaves* e ao final da transmissão o ciclo recomeça.

Para que esse ciclo seja completado corretamente é fundamental que o *framework* seja capaz de escolher frequências onde não ocorram problemas relacionados à interferência, de modo a não causar problemas durante a etapa de transmissão. Para que essa melhor frequência seja utilizada, os métodos de decisão do espectro baseados em aprendizado de máquina são utilizados para definir a melhor frequência em um ambiente específico. Esse métodos são apresentados no capítulo seguinte.

5 MÉTODOS DE DECISÃO BASEADOS EM APRENDIZADO DE MÁQUINA

A utilização de técnicas de aprendizado de máquina para escolha do melhor canal é concentrada na etapa de decisão do espectro. Para que isso seja possível, é necessário criar meios de identificar o comportamento do espectro a fim de prever futuros comportamentos do espectro e desse modo criar maneiras de classificar as faixas do espectro.

A ideia geral por trás de todos os métodos de decisão apresentados nesse capítulo são baseados em informações de tempos passados, e a partir disto tentar identificar o comportamento de cada faixa do espectro em estados futuros, de modo que seja possível prever qual será o estado da rede em um tempo futuro em uma determinada faixa de espectro. Desse modo, será possível avaliar o desempenho de cada faixa do espectro e escolher a faixa que possua o melhor desempenho previsto. Com isso, o *framework* utilizará essa faixa do espectro de forma a maximizar o desempenho da Rede de Rádios Cognitivos.

Os algoritmos de aprendizado de máquina possuem desempenhos diferentes dado o mesmo problema. Dessa forma, a utilização de diferentes métodos de aprendizado de máquina é utilizada neste trabalho para verificar a acurácia de cada um deles dado o modelo proposto. O primeiro método utilizado foi o método do *Random Forest* (RnF), seguido do método *Hidden Markov Model* (HMM).

5.1 *Random Forest*

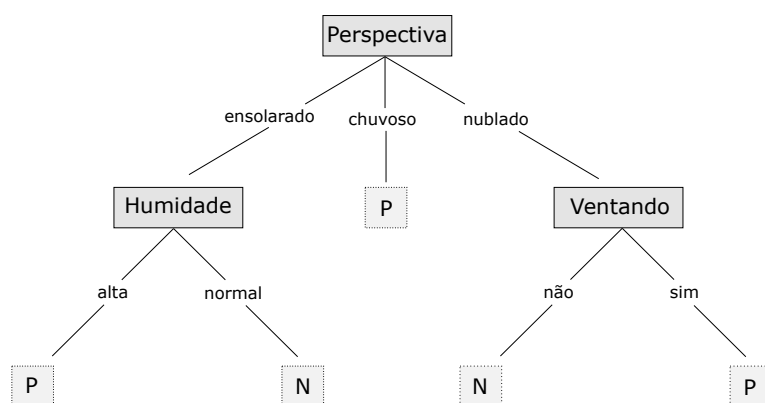
O *Random Forest* é um método de aprendizado de máquina supervisionado, que consiste em muitas ideias simples combinadas para produzir modelos extremamente precisos que podem "aprender" de dados passados. O bloco básico do RnF são as árvores de decisão.

5.1.1 **Árvore de decisão**

Uma árvore de decisão utiliza um gráfico ou modelo semelhante a uma árvore que indica decisões e suas possíveis consequências, como resultados de eventos e custos de recursos. É também uma maneira de exibir graficamente um algoritmo que contém apenas instruções de controle condicional. Uma árvore de decisão é construída utilizando um grafo direcionado $G = (V, E)$, $E \subset V^2$, onde V é o conjunto de vértices e E são as arestas do grafo (KAMIŃSKI; JAKUBCZYK; SZUFEL, 2018).

A Figura 5.1 exemplifica uma árvore de decisão que é desenhada com sua raiz no topo e suas folhas crescendo para baixo, onde P e N denominam instâncias positivas e negativas respectivamente. Os nós que são divididos em duas ou mais partes são chamados de nós intermediários, e quando um nó não pode ser mais dividido o mesmo é denominado nó folha. Uma árvore é denominada árvore de classificação quando representa dados discretos, uma vez que o objetivo é a classificação de um dado em específico. As árvores de regressão por sua vez, são representadas da mesma maneira que as árvores de classificação, porém predizem valores contínuos (QUINLAN, 1986). A criação de uma árvore envolve decidir quais características

Figura 5.1 – Árvore de decisão adaptado de Quinlan (1986)



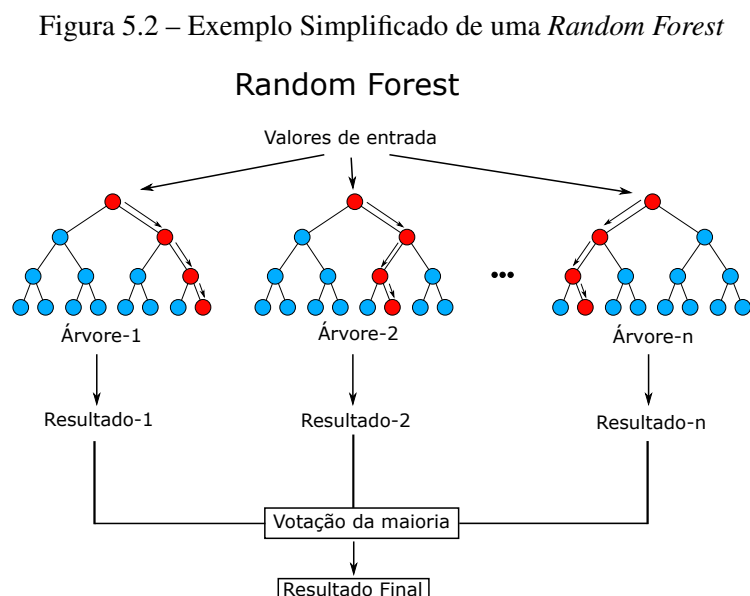
escolher, e quais condições serão usadas para criar a ramificação, além de ter uma condição de parada. Uma técnica para a criação de uma árvore é considerar todos os recursos ou atributos para realizar a primeira divisão da árvore, a divisão na raiz. Todos os dados de treinamento são divididos em grupos e a divisão com o menor custo é determinada como a primeira divisão da árvore. Esse algoritmo é de natureza recursiva, pois os grupos formados podem ser subdivididos utilizando a mesma estratégia. Nessa técnica, todos os recursos são considerados e os pontos de divisão diferentes são testados usando uma função de custo. A divisão com o melhor custo, ou menor custo, dependendo do contexto, é selecionada. Esse algoritmo é conhecido como o algoritmo guloso, pois existe a necessidade de reduzir o custo (QUINLAN, 1983).

Se um problema possui um grande conjunto de recursos, é necessário um grande número de divisões, o que por sua vez, gera uma grande árvore. Essas árvores são complexas e podem levar ao *overfitting*. Por isso, é necessário uma condição de parada. Uma maneira de fazer isso é definindo um número mínimo de entradas de treinamento para usar em cada folha. Uma outra maneira é definir a profundidade máxima da árvore que é o comprimento do caminho mais longo de uma raiz a uma folha.

As vantagens em se utilizar árvores de decisão está na simplicidade em entender, interpretar e visualizar. Além disso, as árvores de decisão executam implicitamente uma seleção de recursos. Além disso, é possível manipular dados numéricos e categorizá-los, além de também poder lidar com problemas de várias saídas. As árvores de decisão exigem de certa forma pouco esforço para a preparação de dados e geralmente produzem bons resultados, mesmo que as suposições definidas sejam de algum modo violadas pelo modelo do qual os dados foram obtidos.

As desvantagens em se utilizar árvores de decisão está na possibilidade de serem criadas de maneira complexas não generalizando bem os dados, causando o chamado *overfitting*. Além disso, não equilibrar o conjunto de dados antes de se ajustar à árvore de decisão pode levar a geração de árvores tendenciosas. Ademais, conceitos como XOR (ou exclusivo), multiplexador ou paridade não são expressados facilmente por árvores de decisão. Além de que, os algoritmos gulosos não podem garantir o retorno da árvore de decisão ótima globalmente. Isso pode ser amenizado pelo treinamento de várias árvores, onde os recursos são amostrados aleatoriamente. O treinamento de diversas árvores é um dos conceitos da *Random Forest*, que será melhor abordado na próxima seção.

A Figura 5.2 apresenta um exemplo de uma *Random Forest* de forma simplificada. Inicialmente, os valores de entrada são passados para cada uma das árvores individualmente, e após a árvore tomar sua decisão, a mesma é adicionada a uma lista com todas as decisões tomadas. É escolhido o resultado que possui o maior número de votos de cada árvore individualmente.



5.1.2 Definição do *Random Forest*

Seja um conjunto de treinamento $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ definidos aleatoriamente a partir de uma distribuição de probabilidade $(x_i, y_i) \sim (X, Y)$. Com o objetivo de construir um classificador que prevê y a partir de x , e que seja baseado no conjunto de dados de D e um conjunto de classificadores $h = \{h_1(x), \dots, h_k(x)\}$. Se cada $h_k(x)$ é uma árvore de decisão, então o seu conjunto é uma *Random Forest* (BREIMAN, 2001).

Os parâmetros da árvore de decisão são definidos para o classificador $h_k(x)$ como $\Theta_k = (\theta_{k1}, \theta_{k2}, \dots, \theta_{kp})$. Esses parâmetros incluem a estrutura da árvore como, por exemplo, quais as variáveis são divididas em determinado nó. Assim, a árvore de decisão k leva a um classificador $h_k(x) = h(x | \Theta_k)$. Os recursos que aparecem no K -ésimo nó da árvore é aleatoriamente definido de acordo com os parâmetros Θ_k , que são definidos aleatoriamente a partir do modelo Θ_k .

Desse modo, uma *Random Forest* é um classificador baseado em um série de classificadores $h(x | \Theta_1), \dots, (x | \Theta_k)$, que são baseados em uma árvore de classificação com parâmetros Θ_k aleatoriamente escolhidos de um modelo Θ . Para a classificação final $f(x)$ (que combina os classificadores), cada árvore vota em uma determinada classe, e a classe com o maior número de votos ganha (BREIMAN, 2001).

A ideia fundamental por trás de uma *Random Forest* é combinar muitas árvores de decisão em um único modelo. Individualmente, as previsões feitas por árvores de decisão podem não ser precisas, contudo, combinadas em conjunto com diversas outras árvores de decisão, as previsões estarão mais próximas da resposta correta.

Cada árvore de decisão na floresta considera um subconjunto aleatório de características e só tem acesso a esse subconjunto aleatório de dados de treinamento. Isso aumenta a diversidade na floresta, levando a previsões gerais mais robustas, e por isto o nome "*Random Forest*". Para fazer uma previsão, a *Random Forest* utiliza o maior número de votos de todas as estimativas das árvores de decisão individuais.

5.2 *Hidden Markov Model*

Uma abordagem diferente é utilizar um processo estocástico que representa o modelo de um conjunto de valores em relação ao tempo. Esta é a ideia do *Hidden Markov Model* - HMM. Essa abordagem trata-se de um sistema de transição de estados, onde o estado atual depende unicamente do estado anterior imediato. O processo discreto de Markov é um processo

estocástico que pode ser chamado de modelo observável de Markov, uma vez que as saídas do processo são um conjunto de estados que correspondem a um evento físico observável. No modelo oculto de Markov a observação é uma função probabilística, dessa maneira, o modelo do processo estocástico não é observável e pode apenas ser observado através de outro conjunto de processos estocásticos que reproduzem a sequência de observações (CHEN; QIU, 2010).

O HMM foi formalmente definido por (RABINER, 1989) e suas características são:

1. N é o número de estados no modelo. Cada estado individual é definido por $S = \{S_1, S_2, \dots, S_n\}$ e o estado no momento t como q_t .
2. M é o número de símbolos de observação distintas por estado. Esses símbolos são denominados

$$V = \{V_1, V_2, \dots, V_m\} \quad (5.1)$$

3. Distribuição de probabilidade de transição de estados $A = \{a_{ij}\}$ onde

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N. \quad (5.2)$$

4. A distribuição de probabilidade do símbolo de observação no estado j , $B = \{b_j(k)\}$ onde

$$b_j(k) = P[v_k | q_t = S_j], \quad 1 \leq j \leq N \text{ e } 1 \leq k \leq M. \quad (5.3)$$

5. A distribuição do estado inicial $\pi = \{\pi_i\}$ onde

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \quad (5.4)$$

Com os valores adequados de N, M, A, B e π , o HMM pode ser então utilizado para gerar uma sequência de observação $O = O_1 O_2 O_3 \dots O_n$, da seguinte maneira:

1. Escolher um estado inicial $q_1 = S_i$ de acordo com a distribuição de estado inicial π .
2. Definir $t = 1$.
3. Escolher $O_t = v_k$ de acordo com a distribuição de probabilidade do símbolo de observação no estado S_i , por exemplo $b_i(k)$.

4. Mudar para novo estado $q_{t+1} = S_j$ de acordo com distribuição de probabilidade de transição de estados.
5. Definir $t = t + 1$, retornar ao passo 3 caso $t < T$ (onde T é o numero de observações na sequência). Caso contrário termine o procedimento.

Esse procedimento pode ser utilizado como um gerador de observações, e como um modelo de como uma dada sequência de observações foi gerada por um HMM adequado. É possível perceber que a especificação de um HMM exige a especificação dos parâmetros de modelo (N e M), dos símbolos de observação e das medidas de probabilidade (A, B e π). Dessa maneira, uma notação compacta para indicar o conjunto de parâmetros do modelo é dada pela equação 5.5.

$$\lambda = (A, B, \pi) \quad (5.5)$$

Rabiner (1989) ainda apresentou três problemas que devem ser solucionados, a fim de que o modelo seja útil em aplicações do mundo real.

1. Dada uma sequência de observação $O = O_1 O_2 O_3 \dots O_T$, e o modelo $\lambda = (A, B, \pi)$, como calcular $P(O|\lambda)$ a probabilidade de uma sequência de observação dado um modelo? A solução para esse problema é a utilização do *forward-backward*, um algoritmo para o cálculo dessa probabilidade (BAUM et al., 1970).
2. Dada uma sequência de observação $O = O_1 O_2 O_3 \dots O_T$, e o modelo $\lambda = (A, B, \pi)$, como escolher a sequência de estados correspondentes $Q = q_1 q_2 \dots q_T$ que melhor explica a sequência de observação? A solução para esse problema é o algoritmo Viterbi (FORNEY, 1973). O Viterbi é um algoritmo de programação dinâmica muito usado para decifrar códigos convolutivos utilizado no protocolo CDMA (*Code Division Multiple Access*) (MILLER; KWONG, 2012), e também em telefonia digital GSM (*Global System Access*) (CYPRIANI et al., 2009).
3. Como determinar um método para ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$, a fim de maximizar a probabilidade da sequência de observação dado o modelo $P(O|\lambda)$? Esse é o problema mais complicado, e de fato não existe solução analítica para esse problema, pois não existe uma solução ótima para estimar o modelo de parâmetros (RABINER, 1989). No entanto, é possível calcular um máximo local utilizando procedimentos iterativos como o método de *Baum-Welch* ou o método de *expectation-modification*

(DEMPSTER; LAIRD; RUBIN, 1977), ou até mesmo utilizando técnicas de gradiente, uma vez que esse problema pode ser definido como um problema de otimização (LEVINSON; RABINER; SONDDHI, 1983).

O segundo modelo que foi adicionado no *framework* foi o *Hidden Markov Model* HMM. Um HMM é um modelo estatístico, no qual o sistema que está sendo modelado é considerado um Processo de Markov com estados ocultos, ou estados não observáveis. Um modelo estatístico estima parâmetros como média e variância e razões de probabilidades de classes a partir dos dados e utiliza esses parâmetros para imitar o que está acontecendo com os dados.

Um processo de Markov é um processo onde a probabilidade de estados futuros do processo depende apenas do estado presente, e não da sequência de estados que o precedem. Esta é a propriedade de Markov e foi definida por Davenport (1970) da seguinte forma.

A probabilidade do processo estocástico X_T em um tempo T estar em um estado do conjunto de estados $S = \{s_1, s_2, \dots, s_n\}$ é apresentado pela Equação 5.6.

$$P(X_t = s_j | X_{t-1} = s_i, X_{t-2} = s_k, \dots) = P(X_t = s_j | X_{t-1} = s_i), \quad s_i, s_j, s_k \in S, t \in T \quad (5.6)$$

Qualquer processo aleatório que satisfaça a propriedade Markov é conhecido como Processo de Markov. Um modelo estatístico que segue o processo de Markov é denominado Modelo de Markov. Existem quatro modelos comuns de Markov, usados em diferentes situações, dependendo se cada estado sequencial é observável ou não, e se o sistema deve ser ajustado com base na observação. A Tabela 7.5 apresenta esses modelos. Dada a natureza dinâmica do espectro, o modelo que se enquadra mais adequadamente com o trabalho proposto é o *Hidden Markov Model*.

Tabela 5.1 – Modelos que utilizam o processo de Markov

Estado	Estados observáveis	Estados parcialmente observáveis
Sistema controlado	<i>Markov Decision Process</i>	<i>Partially observable Markov Decision process</i>
Sistema autônomo	<i>Markov Chain</i>	<i>Hidden Markov Model</i>

5.2.1 Definição do Modelo HMM

A sequência de estado do HMM não é diretamente visível. Só pode ser calculado a partir dos símbolos de observação medidos usando técnicas estatísticas. Um modelo de HMM é definido de forma compacta como $\lambda = (A, B, \pi)$ onde A é a matriz de transição de estados

$$A = [a_{ij}], 1 \leq i, j \leq N \quad (5.7)$$

B é a probabilidade de que a observação O_M ocorra,

$$B = [b_j(k)], 1 \leq j \leq N, 1 \leq k \leq M \quad (5.8)$$

de maneira que cada $b_j(k) = P(y_t = O_k | x_t = s_j), t = 1, 2, \dots$ vale ressaltar que $0 \leq b_j(k) \leq 1$ e ainda que $\sum_{k=1}^M b_j(k) = 1$. E finalmente,

$$\pi = \{\pi_i\} \quad (5.9)$$

onde π_i é a probabilidade do estado inicial de s_i .

Para definir o modelo HMM é necessário conhecer seus parâmetros, contudo, geralmente os parâmetros do modelo de uma situação do mundo real são desconhecidos. Porém, com dados suficientes em mãos é possível fazer essa estimativa desses parâmetros.

Para poder estimar estes parâmetros, primeiramente o modelo é criado com parâmetros aleatórios, que serão eventualmente treinados para corresponder aos dados reais. O modelo será treinado alterando os valores dos parâmetros, baseados nas várias sequências de variáveis de saída, ou emissões. As probabilidades do modelo estimado são semelhantes ao modelo real, mas caso não sejam, o procedimento de estimativa é novamente executado por mais tempo e em diferentes modelos, gerados aleatoriamente, pois existe a possibilidade da estimativa, dado um estado inicial, cair em um ótimo local e não no ótimo global. Para verificar se as probabilidades do modelo estimado refletem o modelo real, uma sequência de estados de saída é gerada, caso a sequência gerada coincida com a sequência utilizada na etapa de treinamento, o treinamento foi completado com sucesso, e o modelo estimado reflete o modelo real. O algoritmo Baum Welch é o algoritmo utilizado para estimar os parâmetros desconhecidos do HMM (JUANG; RABINER, 1985).

5.2.2 Baum Welch

O algoritmo Baum Welch pode ser utilizado para estimar dos parâmetros desconhecidos de um HMM, dado um conjunto de características observadas (BILMES, 1998).

Um HMM X_t com N valores possíveis que possua matriz de transição $A = \{a_{ij}\} = P(X_t = j | X_{t-1} = i)$ e com estados iniciais $\pi_i = P(X_1 = i)$. As variáveis de saída ou observações Y_t podem assumir K possíveis valores. A probabilidade de uma observação y_i em um tempo t para o estado $X_t = j$ é dado por $b_j(y_i) = P(Y_t = y_i | X_t = j)$. Levando em consideração todos os valores possíveis de Y_t e X_t obtém-se a matriz $B = \{b_j(y_i)\}$ de tamanho $N \times K$ onde b_j são os possíveis estados e y_i são as observações. A sequência de observação é dada por $Y = (Y_1 = y_1, Y_2 = y_2, \dots, Y_T = y_T)$. Dado o modelo HMM $\lambda = (A, B, \pi)$ o algoritmo de Baum Welch encontra $\lambda' = \arg \max_{\lambda} P(Y | \lambda)$.

Primeiramente o algoritmo calcula recursivamente a probabilidade de observar os estados y_1, y_2, \dots, y_t e estar no estado i no momento t , ou seja $\alpha_i(t) = P(Y_1 = y_1, \dots, Y_t = y_t, X_t = i | \theta)$.

$$\begin{aligned} \alpha_i(1) &= \pi_i b_i(y_1), \\ \alpha_i(t+1) &= b_i(y_{t+1}) \sum_{j=1}^N \alpha_j(t) a_{ji} \end{aligned} \quad (5.10)$$

Em seguida, calcula recursivamente a probabilidade acabar a sequência parcial y_{t+1}, \dots, y_T dado o estado inicial i no momento t , ou seja

$$\begin{aligned} \beta_i(T) &= 1, \\ \beta_i(t) &= \sum_{j=1}^N \beta_j(t+1) a_{ij} b_j(y_{t+1}) \end{aligned} \quad (5.11)$$

As variáveis temporárias podem ser agora calculadas de acordo com o teorema de Bayes (BAYES; PRICE, 1763)

$$\gamma_i(t) = P(X_t = i | Y, \theta) = \frac{P(X_t = i, Y | \theta)}{P(Y | \theta)} = \frac{\alpha_i(t) \beta_i(t)}{\sum_{j=1}^N \alpha_j(t) \beta_j(t)} \quad (5.12)$$

A Equação 5.12 é a probabilidade de estar no estado i no momento t dada a sequência parcial de observação Y e os parâmetros do modelo λ

$$\xi_{ij}(t) = P(X_t = i, X_{t+1} = j | Y, \theta) = \frac{P(X_t = i, X_{t+1} = j, Y | \theta)}{P(Y | \theta)} = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})} \quad (5.13)$$

A Equação 5.13 é a probabilidade de estar no estado i e j no momento t e $t + 1$ respectivamente, dado a sequência de observação Y e os parâmetros do modelo λ .

Desse modo, os parâmetros do modelo λ serão atualizados. Os estados iniciais serão $\pi'_i = \gamma_i(1)$. A matriz de transições é dada pela Equação 5.14.

$$a'_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \quad (5.14)$$

A matriz de emissão é dada por uma função, onde $b_i^*(v_k)$ é o número esperado de vezes que as observações de saída foram iguais para v_k enquanto estiver no estado i sobre o número total de vezes esperado no estado i .

$$b_i^*(v_k) = \frac{\sum_{t=1}^T 1_{y_t=v_k} \gamma_i(t)}{\sum_{t=1}^T \gamma_i(t)} \quad 1_{y_t=v_k} = \begin{cases} 1, & \text{se } y_t = v_k, \\ 0, & \text{se diferente} \end{cases} \quad (5.15)$$

Essas etapas são repetidas até que converjam, e assim os parâmetros de um HMM podem ser estimados e um modelo HMM adequadamente treinado. Uma vez que os parâmetros do modelo são conhecidos, é possível então gerar uma sequência de estados ocultos de modo a prever os estados futuros não observáveis.

5.2.3 Algoritmo de Viterbi

O algoritmo Viterbi é um algoritmo de programação dinâmica utilizado com a finalidade de encontrar a sequência mais provável de estados ocultos. O algoritmo foi proposto por Andrew Viterbi em 1967, como um algoritmo de decodificação de códigos convolucionais sobre links de comunicação digital com grande interferência (FORNEY, 1973).

Como explicado por Viterbi (1967), o algoritmo gera um vetor $X = (x_1, x_2, \dots, x_T)$, que é uma sequência de estados $x_n \in S = \{s_1, s_2, \dots, s_K\}$ que irá gerar as observações $Y = (y_1, y_2, \dots, y_T)$ de maneira que $y_n \in O = \{o_1, o_2, \dots, o_N\}$ onde N é a quantidade de observações do espaço de observações. Os parâmetros de entrada do algoritmo são compostos por O, S, Π, Y, A, B , onde

- O é o espaço de observação $O = (o_1, o_2, \dots, o_N)$.
- S é o espaço de estados, ou conjunto de todas as configurações ou estados possíveis de um sistema, $S = \{s_1, s_2, \dots, s_K\}$.

- Π é uma matriz de probabilidades iniciais $\Pi = (\pi_1, \pi_2, \dots, \pi_K)$ de maneira que π_i armazena a probabilidade de $x_1 = s_i$.
- Y é uma sequência de observações $Y = (y_1, y_2, \dots, y_T)$ de tal modo que $y_t = i$.
- A é a matriz de transições A de tamanho $K \times K$ tal que A_{ij} armazena a probabilidade de transição do estado s_i para s_j .
- B é a matriz de emissão de tamanho $K \times N$ tal que B_{ij} armazena a probabilidade de observar o_j pelo estado s_i .

A saída é a sequência de estados $X = (x_1, x_2, \dots, x_N)$.

Algorithm 1 Algoritmo Viterbi

```

function VITERBI( $O, S, \Pi, Y, A, B$ ): $X$ 
  for each state  $i \in \{1, 2, \dots, K\}$  do
     $T_1[i, j] \leftarrow \pi_i \cdot B_{ij1}$ 
     $T_2[i, j] \leftarrow 0$ 
  end for
  for each observation  $i = 2, 3, \dots, T$  do
    for each state  $j \in \{1, 2, \dots, K\}$  do
       $T_1[j, i] \leftarrow \max_k (T_1[k, i-1] \cdot A_{kj} \cdot B_{jy_i})$ 
       $T_2[j, i] \leftarrow \arg \max_k (T_1[k, i-1] \cdot A_{kj} \cdot B_{jy_i})$ 
    end for
  end for
   $z_T \leftarrow \arg \max_k (T_1[k, T])$ 
   $x_T \leftarrow S_{z_T}$ 
  for  $i \leftarrow T, T-1, \dots, 2$  do
     $z_{i-1} \leftarrow T_2[z_i, i]$ 
     $x_{i-1} \leftarrow S_{z_{i-1}}$ 
  end for
  return  $X$ 
end function

```

O algoritmo Viterbi apresentado em Forney (1973) pode ser visualizado no Algoritmo 1. Um HMM com espaço de estados S , probabilidade de estados iniciais π_i , probabilidade de transição de estados $a_{i,j}$ de mudar do estado i para o estado j e eventos observáveis y_1, \dots, y_T . A provável sequência de estados observáveis x_1, \dots, x_T é dada pela relação de recorrência da Equação 5.16.

$$\begin{aligned}
 V_{1,k} &= P(y_1 | k) \cdot \pi_k \\
 V_{t,k} &= \max_{x \in S} (P(y_t | k) \cdot a_{x,k} \cdot V_{t-1,x})
 \end{aligned}
 \tag{5.16}$$

Assim $V_{t,k}$ é a probabilidade da sequência de estados que é dada por $P(x_1, \dots, x_t, y_1, \dots, y_t)$ das t observações que possuem como o estado final k . A sequência de observações encontradas pelo algoritmo Viterbi são obtidas armazenando o estado observável x . Assim, seja $Ptr(k, t)$ a função que retorna o valor de x utilizado para calcular $V_{t,k}$ se $t > 1$ ou k se $t = 1$ então

$$\begin{aligned} x_T &= \arg \max_{x \in S} (V_{T,x}) \\ x_{t-1} &= Ptr(x_t, t) \end{aligned} \quad (5.17)$$

Arg max são os pontos do domínio da função, na qual os valores são maximizados. Resumidamente, o algoritmo acompanha o estado com a maior probabilidade em cada estágio. No final da sequência, o algoritmo irá verificar todos os estados, selecionando o estado com maior probabilidade a cada passo de tempo e, assim, criando a provável sequência de estados ocultos que levam à sequência de observações.

5.3 Rede Neural Artificial

As Redes Neurais Artificiais são sistemas paralelos distribuídos compostos por nodos, que são unidades de processamentos simples. Esses nodos executam determinadas funções matemáticas e são dispostos em uma ou mais camadas e interligadas por um grande número de conexões. Na maioria dos modelos estas conexões estão associadas a pesos, e esses pesos armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. As RNAs são inspiradas pela estrutura física concebida pela natureza do cérebro humano.

O procedimento utilizado na solução de problemas nas RNA passa inicialmente por uma fase de aprendizagem, onde um conjunto de exemplos é apresentado para rede que extrai automaticamente as características necessárias para representar a informação fornecida. Essas informações são utilizadas posteriormente para gerar respostas para o problema.

O modelo de um neurônio artificial é uma simplificação do neurônio biológico. O neurônio artificial possui n terminais de entrada que representam os dendritos do neurônio biológico, e uma saída y que representa o axônio do neurônio biológico. Para representar as sinapses do neurônio biológico as entradas possuem pesos w_1, \dots, w_n , que são valores positivos ou negativos, dependendo do treinamento realizado no neurônio. O neurônio executa quando ele recebe um impulso que ultrapassada um *threshold*, essa ativação é dada por uma função de ativação da

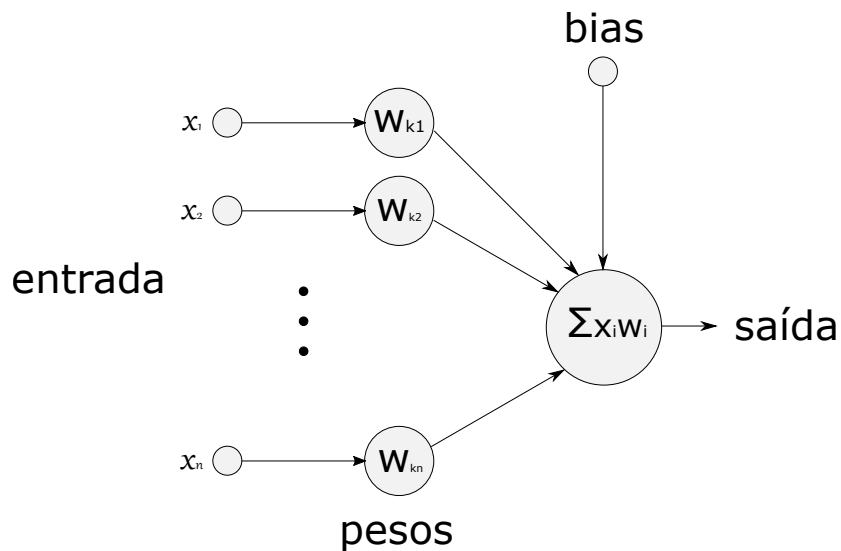
Equação 5.18 definida por McCulloch e Pitts (1943) para o nodo MCP, onde n é o número de entradas do neurônio, w_i é o peso associado à entrada x_i e θ é o *threshold*.

$$\sum_{i=1}^n x_i w_i \geq \theta \quad (5.18)$$

Esse trabalho serviu de base para a modelagem do perceptron que é uma evolução da modelagem do nodo. O modelo proposto por Rosenblatt (1958), conhecido como perceptron, era composto por uma estrutura de rede tendo como unidades básicas nodos MCP e uma regra de aprendizado. A estrutura de um perceptron é apresentada pela Figura 5.3.

O algoritmo de aprendizagem do perceptron utiliza o algoritmo de correção de erros. Após o processamento de uma dada entrada para o perceptron o seu resultado é verificado. Caso o valor seja incorreto é aplicado o algoritmo de treinamento onde os pesos serão alterados. Após a atualização dos pesos, a entrada é novamente aplicada ao perceptron, caso o valor esteja correto não será necessário a atualização dos pesos, porém se estiver novamente incorreto o algoritmo de treinamento é novamente aplicado, até que seja obtido um valor aceitável.

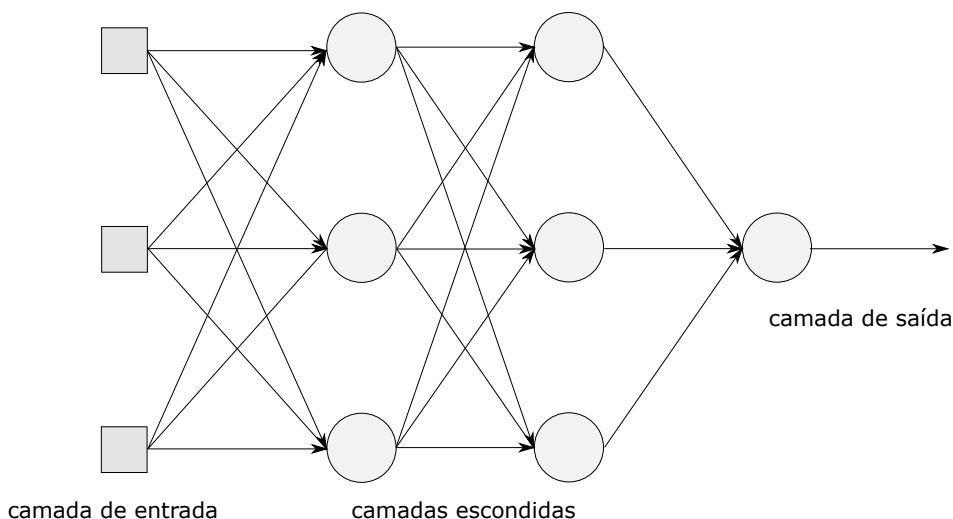
Figura 5.3 – Perceptron de uma camada adaptado de Rosenblatt (1958)



Para resolver problemas complexos uma rede de diversas camadas de neurônios deve ser utilizada. De acordo com Cybenko (1989), uma rede que possua uma camada intermediária pode implementar qualquer função contínua e a utilização de duas camadas intermediárias permite a aproximação de qualquer função. As *MultiLayer Perceptron* (MLP) são RNAs que utilizam pelo menos uma camada intermediária ou escondida.

As MLP apresentam um poder computacional maior do que redes sem camadas intermediárias, pois elas conseguem tratar dados que não são linearmente separáveis. A precisão obtida e a implementação de uma rede MLP dependem do número de nodos utilizados nas camadas intermediárias. Na Figura 5.4 é apresentada um rede MLP típica.

Figura 5.4 – Rede MLP com duas camadas intermediárias



A definição de uma topologia de uma rede MLP está ligada com a definição dos nodos da camada intermediária. Este número é, geralmente, definido empiricamente e depende da distribuição dos padrões de treinamento e validação da rede. O algoritmo de aprendizado mais conhecido para treinamento destas redes é o *back-propagation* (RUMELHART; MCCLELLAND; University of California, 1986).

O *back-propagation* é baseado na regra proposta por Widrow e Hoff (1960). Este algoritmo propõe uma forma de definir o erro dos nodos das camadas intermediárias, possibilitando o ajuste de seus pesos. O *back-propagation* é um algoritmo supervisionado, que utiliza pares (entrada, saída desejada) para, através de um mecanismo de correção de erros, ajustar os pesos da rede. O treinamento ocorre em duas fases, onde cada fase percorre a rede em um sentido. Estas duas fases são chamadas de fase *forward* e fase *backward*. O algoritmo define para cada padrão de treinamento uma saída da rede através da fase *forward*. Esta saída produzida é então comparada com a saída desejada, e os pesos dos nodos são atualizados através da fase *backward*. Isso ocorre até que o erro da saída desejada com a produzida seja mínimo.

6 PROTOCOLOS DE ROTEAMENTO

A maior parte das pesquisas relacionadas as Redes de Rádios Cognitivos até o presente momento concentraram-se em cenários de um único salto. Estas pesquisas abordaram questões relacionadas apenas as camadas física e MAC sobre o sensoriamento do espectro, decisão do espectro e compartilhamento do espectro (CORMIO; CHOWDHURY, 2009). Recentemente, contudo, a comunidade científica começou a perceber o potencial das RRC multi-hop. O paradigma dos Rádios Cognitivos pode ser utilizado em diferentes cenários multi-hop, como redes Mesh e redes Ad Hoc de Rádios Cognitivos (CHOWDHURY; AKYILDIZ, 2008). Alguns trabalhos começaram a abordar os desafios dos protocolos de roteamento para as Redes de Rádios Cognitivos, e até mesmo a propor possíveis soluções (CESANA; CUOMO; EKICI, 2011).

Visando acrescentar a comunicação multi-hop no *framework* proposto, para que o mesmo seja capaz de comunicar a distâncias maiores sem necessariamente utilizar o aumento da potência de transmissão dos nós, foram implementados dois algoritmos de roteamento, o AODV e OLSR. Esses protocolos foram escolhidos pois o AODV é a base de desenvolvimento de muitos protocolos encontrados na literatura, enquanto que o OLSR foi escolhido por ser um protocolo distribuído e otimizado, apesar de existirem diversas variações deste protocolo.

6.1 *Ad Hoc On-Demand Distance Vector*

O protocolo de roteamento AODV (*Ad Hoc On-Demand Distance Vector*) foi projetado para redes sem fio e redes móveis ad hoc, que oferece rápida adaptação às condições de enlace dinâmico, baixo custo de processamento e memória, baixa utilização de rede e a determinação de rotas *unicast* para destinos dentro da rede ad hoc. O protocolo AODV foi desenvolvido em conjunto pelo Nokia Research Center, a Universidade da Califórnia, em Santa Bárbara e a Universidade de Cincinnati, no ano de 2003 (PERKINS; BELDING-ROYER; DAS, 2003). O AODV é um algoritmo reativo, de forma que são criadas rotas entre nós apenas se for solicitado pelos nós de origem, por isso, esse protocolo é considerado um algoritmo sob demanda e não cria nenhum tráfego extra para comunicação ao longo dos links. As rotas são mantidas por todos os nós que tenham conhecimento da comunicação entre origem e destino.

A rede se mantém em silêncio até que seja necessária o estabelecimento de uma conexão. O nó da rede que precisa de conexão transmite uma solicitação de conexão em *broadcast*. Em seguida, os nós da rede encaminham esta mensagem para seus vizinhos, ao mesmo tempo em que registram o nó que solicitou a conexão. Assim, são criadas rotas para a resposta ao nó

solicitante. Um nó intermediário que recebe tais mensagens, mantém uma rota para um nó desejado, e envia uma mensagem de retorno por meio de rotas armazenadas para o nó solicitante. E através disso, são criadas uma série de rotas de volta ao nó solicitante. O nó que iniciou a solicitação usa a rota que contém o menor número de saltos através dos nós da rede.

O protocolo define tipos de mensagens, como solicitações de rota (*Route Requests* - RREQs), assim como respostas de rota (*Route Replies* - RREPs). O AODV não realiza nenhuma operação quando os nós de uma rede possuem rotas válidas para a comunicação entre si. Conseqüentemente, quando uma nova rota para um nó se faz necessária, o nó que deseja realizar a transmissão envia um RREQ, para que seja encontrada uma rota para o destino. Uma rota pode ser determinada quando o RREQ alcança o nó destino ou um outro nó intermediário, com uma rota válida para o destino. A rota é disponibilizada e um RREP é enviado de volta para a origem do RREQ. Cada nó que recebe a solicitação armazena uma rota de volta para o originador da solicitação, de maneira que o RREP possa encontrar uma rota entre o nó de origem e destino ou, da mesma maneira, de qualquer nó intermediário que seja capaz de satisfazer a solicitação. O AODV é um protocolo de roteamento e por esse motivo deve lidar com o gerenciamento de tabelas de rotas. O protocolo utiliza os campos de endereço de destino, próximo salto e contagem de saltos em sua tabela de rotas.

Operação do Protocolo

Esta seção descreve os cenários sob os quais os nós geram mensagens de requisição de rota RREQ, e resposta de rota RREP, para comunicação unicast, em direção a um destino e como os dados da mensagem são tratados. Para processar as mensagens corretamente, determinadas informações de estado devem ser mantidas nas entradas da tabela de rotas para os destinos de interesse.

Após um nó receber um pacote de um vizinho, ele verifica sua tabela de rotas, uma entrada para o nó de destino. Caso não exista uma entrada correspondente para esse destino, uma nova entrada é criada. Uma rota já existente na tabela, só é atualizada se a contagem de saltos do pacote contendo a nova informação mais um, for menor que a contagem de saltos existente.

Criando Requisição de Rotas - RREQ

Um nó requisita um RREQ quando determina a necessidade de uma rota para um nó destino, e a mesma não está disponível. Isso pode acontecer se o nó de destino for previamente desconhecido para o nó, ou se ocorrer uma desconexão na rede. O campo Contagem de saltos é definido para zero. Ao receber um RREQ o nó de origem verifica no RREQ o endereço do originador (seu próprio endereço). Desta forma, quando o nó recebe o pacote novamente de seus vizinhos, ele não reprocessará e redirecionará o mesmo pacote.

Um nó de origem deseja que haja uma comunicação bidirecional com o nó de destino. Por esse motivo, apenas uma rota do nó de origem para o nó de destino não é suficiente, o destino necessita também que exista uma rota conhecida para o nó de origem. Para que isso aconteça, qualquer geração de um RREP por um nó intermediário para entrega ao nó de origem, precisa também de alguma ação que notifique o destino sobre uma rota de volta para o nó de origem.

Processando e encaminhando solicitações de rota

Primeiramente é incrementado o valor da contagem de saltos no RREQ em um, para considerar o novo salto através do nó intermediário. Em seguida, o nó procura na tabela uma rota inversa para o endereço do nó de origem. Se for necessário, a rota é criada ou atualizada utilizando o número de saltos do nó de origem do RREQ em sua tabela de roteamento. Quando a rota inversa é criada ou atualizada, as seguintes ações são executadas:

Os números de saltos do nó de origem do RREQ é comparado com o número de saltos do nó de destino na entrada da tabela de roteamento e será copiado se for menor que o valor na tabela de roteamento. Em seguida, o próximo salto na tabela de roteamento se torna o nó do qual o RREQ foi recebido, essa informação é obtida do endereço de origem do pacote recebido, e geralmente é diferente do endereço do nó originador da mensagem RREQ. Por fim, a contagem de saltos atualizada é copiada na mensagem RREQ. Dessa maneira, é criada uma rota inversa para o nó de origem, e o nó atual pode usar a rota inversa para encaminhar pacotes de dados, da mesma maneira que para qualquer outra rota na tabela de roteamento.

Respostas de rota - RREP

Um nó gera um RREP se ele é o próprio nó de destino, ou se conhece uma rota para o nó de destino. Uma vez criado, o RREP é enviado para o próximo salto em direção ao nó de origem do RREQ, de acordo com a entrada da tabela de rotas. Conforme o RREP é encaminhado de

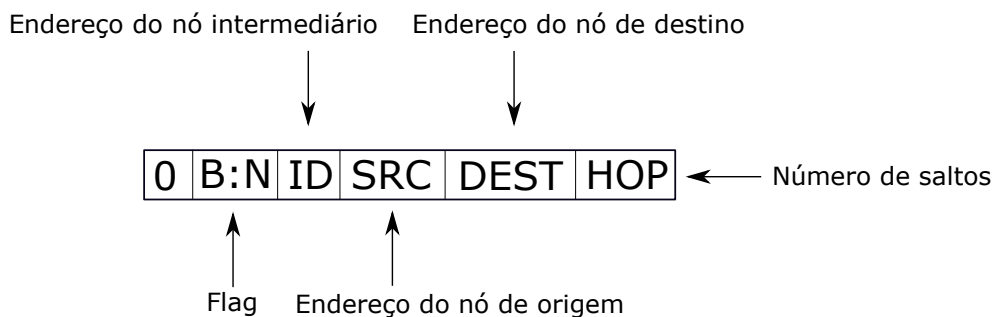
volta para o nó que originou a mensagem RREQ, a contagem de saltos é incrementada a cada salto. Dessa maneira, quando o RREP alcança o nó de origem, a contagem de saltos será a distância da origem ao destino.

Quando um nó recebe uma mensagem RREP, ele procura por uma rota para o salto anterior. Em seguida, o nó incrementa o valor da contagem de saltos, para considerar o novo salto através do nó intermediário. Assim, a rota de encaminhamento para esse destino é criada. A entrada existente na tabela é atualizada somente se a nova contagem de saltos for menor do que a contagem de saltos da tabela. O nó atual pode usar essa rota para encaminhar pacotes de dados para o destino. Caso o nó atual não seja o nó de destino da mensagem RREP, o nó consultará sua tabela de roteamento, para determinar o próximo salto do encaminhamento do RREP, e em seguida encaminhar o RREP usando as informações da tabela de rotas. Caso nenhum RREP gerado a partir da mesma tentativa de descoberta de rota atinja o nó que originou a mensagem RREQ, o nó originador tentará novamente a descoberta da rota após um tempo limite.

Formato das Mensagens

Para a implementação do AODV no *framework* o formato das mensagens teve que ser adaptado para que pudesse ser incluído com a atual estrutura do *framework* proposto. O formato do pacote utilizado para mensagem de RREP (solicitação de rota) e RREQ (resposta de rota) é ilustrado pela Figura 6.1.

Figura 6.1 – Formato do pacote de mensagens de controle



O primeiro campo possui o valor 0 (zero), este é um indicador de que esse pacote pertence a essa Rede de Rádios Cognitivos. O segundo campo é o campo *flag* que pode conter o valor B ou N. O valor B no campo de *flag* é utilizado para indicar uma solicitação de rota, enquanto o valor N é utilizado para indicar uma resposta de uma solicitação de rota. O campo ID indica o endereço do nó que manipula a solicitação, todos os endereços utilizados nos campos

relacionados a endereço do nó são compostos por identificadores únicos dos nós na rede, esse campo é modificado em cada nó antes de realizar o encaminhar o pacote de controle. O campo SRC indica o endereço do nó que originou o pedido de Rota. O campo DEST indica endereço do destino para o qual uma rota é desejada. Finalmente, o campo HOP indica o número de saltos do endereço do originador para o nó que manipula a solicitação.

Neste *framework* todos os nós conhecem a maior parte de seus vizinhos após a etapa de descoberta de vizinhos. Caso um nó deseje enviar um pacote para algum outro que não esteja em sua tabela de rotas, o procedimento do AODV é utilizado e será enviada a mensagem de RREQ para a rede a fim de encontrar a rota para o nó desejado. Esta mensagem será respondida com uma mensagem RREP e a rota entre origem e destino é encontrada. Caso haja a desconexão de um nó na rede, e o protocolo não consiga encontrar uma rota para o nó desejado, possivelmente por esse nó estar usando uma frequência diferente dos outros nós da rede, o ciclo de execução do *framework* é reiniciado.

6.2 *Optimized Link State Routing Protocol*

O segundo protocolo de roteamento implementado nesse *framework* foi o OLSR. O *Optimized Link State Routing Protocol* (OLSR), é um protocolo de roteamento pró-ativo orientado por tabela, desenvolvido para redes móveis densas (CLAUSEN; JACQUET, 2003). Ele é uma otimização de protocolos de estado de enlace puro, pois reduz o tamanho do pacote de controle, bem como o número de transmissão de pacotes de controle necessários. O OLSR reduz a sobrecarga no tráfego de controle utilizando *Multipoint Relays* (MPR), sendo esse o ponto principal do protocolo OLSR. Um MPR é um nó que foi escolhido entre seus vizinhos de um salto para encaminhar pacotes. Ao invés de todos os nós da rede encaminharem pacotes, apenas os nós MPR fazem esse encaminhamento dos pacotes. Isso reduz a sobrecarga de rede, por isso, é mais eficiente do que os protocolos de roteamento de estado de link puro.

O OLSR é ideal para redes móveis grandes e densas. Devido ao uso de MPRs, quanto maior e mais densa for a rede, mais otimizado será o roteamento, fornecendo o caminho mais curto para um determinado destino. O único requisito é que todos os MPRs declarem as informações de link para seus nós vizinhos de MPR. As informações de topologia de rede são mantidas periodicamente por informações de estado de link de troca. Se for necessária mais reatividade a alterações topológicas, o intervalo de tempo para troca de informações de estado de link pode ser reduzido.

Multipoint Relays

A ideia de *Multipoint Relays* é minimizar a sobrecarga de mensagens na rede, reduzindo as retransmissões redundantes em uma mesma região. Cada nó da rede seleciona um conjunto de nós em sua vizinhança de primeiro salto, que pode retransmitir suas mensagens. Este conjunto de nós vizinhos selecionados é chamado de "*Multipoint Relays*"(MPR) desse nó. Os vizinhos do nó N que não estão em seu conjunto de MPR, recebem e processam mensagens de difusão, mas não retransmitem as mensagens recebidas do nó N. Cada nó seleciona seu conjunto de MPR entre seus vizinhos de um salto. Este conjunto é selecionado de tal forma que cobre todos os nós em termos de alcance de rádio.

Os MPRs são usados para controlar as mensagens de inundação de um nó na rede, reduzindo o número de retransmissões. Assim, o conceito de MPR é uma otimização de um mecanismo clássico de inundação. Cada nó da rede seleciona, independentemente, seu próprio conjunto de MPRs entre seus vizinhos. O conjunto de MPR deve ser recalculado quando forem detectadas alterações na vizinhança.

Operação do Protocolo

Neste *framework* o protocolo inicia com uma descoberta de vizinhos enviando uma mensagem em *broadcast*. Os nós vizinhos que recebem essa mensagem replicam o *broadcast* para os próximos nós e em seguida respondem a descoberta de vizinhos para o nó solicitante. As informações do nós como número de saltos e potência de transmissão do nó são armazenadas na tabela de roteamento, de modo que as entradas da tabela de roteamento são compostas por endereço do nó de destino, endereço do nó para o próximo salto, contagem de saltos e força do sinal em dbm do próximo salto. Quando um nó envia um pacote qualquer, a força do sinal é obtida e armazenada para esse nó.

Tabela 6.1 – Formato de mensagens OLSR

Tipo	Formato
Solicitação de Descoberta de vizinhos	<0:B:ID:HOP>
Resposta descoberta de vizinhos	<0:N:ID:SID:HOP>

A tabela 6.1 apresenta o formato das mensagens utilizadas para a descoberta de vizinhos, onde é enviada a mensagem indicando a solicitação da descoberta de vizinhos pelo campo B. O campo ID indica o endereço do nó que realizou a solicitação da DV e o campo HOP indica o número de saltos para o nó que solicitou a descoberta de vizinhos. Após a solicitação da

descoberta de vizinhos, o nó responde com a mensagem indicando a resposta da descoberta de vizinhos pelo campo N. O campo ID indica o endereço do nó que está manipulando a mensagem, uma vez que pode ser um nó intermediário. O campo SID indica o endereço do nó que respondeu a solicitação da descoberta de vizinhos, e por fim, o campo HOP indica o número de saltos para o nó que solicitou a descoberta de vizinhos. Cada nó armazena as informações do número de saltos para os nós da rede em sua tabela de roteamento juntamente com o RSSI do nó que enviou a mensagem. Dessa maneira, além do número de saltos para um determinado nó ainda é definido o RSSI. Com essa informação adicional do RSSI é feita a escolha do MPR. O nó com RSSI mais alto será o nó que escolhido para encaminhar as mensagens. Caso o nó que foi escolhido como MPR se desconecte da rede, um outro nó será escolhido para encaminhar os pacotes, esse nó será o que possuir o RSSI no momento. Caso não exista nós na rede para substituir o MPR que foi desconectado da rede, o *framework* reinicia o seu ciclo de execução.

7 METODOLOGIA DE DESENVOLVIMENTO E RESULTADOS

Esta seção apresenta a metodologia de desenvolvimento utilizada para a avaliação do trabalho, a maneira que foram definidos os modelos dos métodos de aprendizado de máquina, as topologias utilizadas para a realização dos teste, as métricas de avaliação e os resultados obtidos.

7.1 Modelagem dos métodos de aprendizado de máquina

Os algoritmos de aprendizado de máquina são capazes de manipular uma certa quantidade de dados, com a finalidade de encontrar padrões para construir uma série de regras de decisão, para então realizar uma classificação desses dados. Na implementação de algoritmos de aprendizado de máquina, grande parte do tempo gasto na análise de dados está na limpeza e refinamento dos dados. Além disso, o objetivo dos modelos propostos está em realizar a previsão do estado futuro de uma determinada frequência com a finalidade de avaliar o seu desempenho nesse estado futuro.

Inicialmente são necessários dados para serem passados como valores de entrada para os modelos. Por isso, foi realizada uma varredura do espectro a partir da frequência de 800MHz a 5,9GHz utilizando uma USRP B200 da Ettus Research, onde foram capturados os níveis de ruído em cada faixa de frequência. Essa varredura foi realizada por um período de sete dias e gerou um grande volume de dados brutos. Os dados brutos são compostos pela data e hora, a faixa de frequência analisada e o ruído base. Uma amostra dos dados brutos pode ser vista pela Tabela 7.1.

Tabela 7.1 – Exemplo de uma amostra dos dados brutos coletados

Data e Hora da coleta	Faixa de Frequência	Ruído Base
2015-02-06 15:01:37.100214	2398750000.0 - 2398637500.0	-108.135159771
2015-02-06 15:01:39.347806	2401250000.0 - 2401137500.0	-102.557844511
2015-02-06 15:01:41.598915	2403750000.0 - 2403637500.0	-84.7250684735
2015-02-06 15:01:43.850369	2406250000.0 - 2406137500.0	-80.3144946976
2015-02-06 15:01:46.100940	2408750000.0 - 2408637500.0	-72.1765465204
2015-02-06 15:01:48.347979	2411250000.0 - 2411137500.0	-68.5764761708
2015-02-06 15:01:50.598741	2413750000.0 - 2413637500.0	-69.9137826525

Infelizmente, ainda não é possível apenas enviar dados brutos para um modelo e fazer com que ele retorne uma resposta satisfatória, assim alguma quantidade de manipulação de

dados é necessária para qualquer aplicação de aprendizado de máquina. Primeiramente, foi necessário realizar pequenas modificações para colocar os dados em termos compreensíveis à máquina. Por isso, os dados brutos foram modificados de maneira a utilizar as características da hora sem os minutos, frequência em GHz e ruído base em dBm. Uma amostra dos dados pré-processados pode ser vista pela Tabela 7.2.

Tabela 7.2 – Exemplo de uma amostra dos dados pré-processados

Hora	Frequência	Ruído Base
14	2.401	-100
14	2.403	-102
14	2.406	-84
14	2.408	-80
14	2.411	-72
14	2.413	-68
14	2.416	-69

Após a modificação dos dados brutos, os dados modificados podem então ser passados como parâmetros de entrada para os métodos de aprendizado de máquina. De posse desses dados, cada método de aprendizado de máquina pode então executar suas previsões e chegar a alguma solução de acordo com a modelagem realizada.

7.1.1 *Random Forest*

O método *Random Forest* (RnF) é um aprendizado de máquina supervisionado, que consiste em muitas ideias simples combinadas para produzir modelos extremamente precisos, que podem ser treinados para identificar padrões.

Uma vez que foram definidos os dados que serão utilizados pelo RnF, esses dados devem ser separados em *features* e objetivos. O objetivo é o valor a ser previsto, nesse caso, a classe da frequência, enquanto as *features* são todas as demais informações que o modelo usa para realizar uma previsão.

As classes das frequências foram definidas a partir do ruído base de uma frequência específica, no momento em que a mesma foi analisada. Quanto maior o sinal amostrado, maior será o ruído naquela frequência. Desse modo, as classes foram definidas tendo como base o valor do ruído na frequência amostrada, assim, as classes foram divididas em 6 grupos que são apresentados na Tabela 7.3.

Tabela 7.3 – Classificação da frequência em relação ao ruído base

Classificação	Qualidade	Faixa
1	Péssimo	Acima de -85
2	Ruim	-85 a -90
3	Regular	-90 a -95
4	Bom	-95 a -100
5	Ótimo	-100 a -120
6	Excelente	-120 e abaixo

Uma vez que a classificação das frequências foram definidas, foi realizado então a classificação das amostras após o tratamento dos dados brutos. Essa classificação foi estabelecida e pode analisada pela Tabela 7.4. Esta classificação leva em consideração tanto o estado da frequência atual quanto a previsão do estado da frequência na próxima hora.

Tabela 7.4 – Definição das classes para a classificação dos dados das amostras

Classificação	Descrição
1	Sinal péssimo e péssimo na próxima hora
2	Sinal péssimo e melhor ou o mesmo na próxima hora
3	Sinal ruim e pior na próxima hora
4	Sinal ruim e melhor ou o mesmo na próxima hora
5	Sinal regular e pior na próxima hora
6	Sinal regular e melhor ou o mesmo na próxima hora
7	Sinal bom e pior na próxima hora
8	Sinal bom e melhor ou mesmo na próxima hora
9	Sinal ótimo e pior na próxima hora
10	Sinal ótimo e melhor ou mesmo na próxima hora
11	Sinal excelente e pior na próxima hora
12	Melhor caso, sinal excelente e excelente na próxima hora

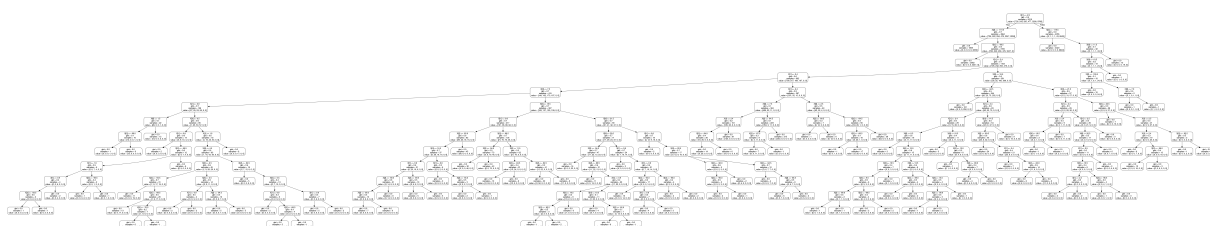
A etapa final da preparação dos dados é a divisão dos dados em conjuntos de treinamento e testes. Durante a etapa do treinamento da *Random Forest*, o modelo analisa as classes para que desse modo possa aprender a prever o resultado a partir das *features* disponibilizadas. Espera-se que haja uma relação entre as *features* e o valor desejado (classe), e o objetivo do modelo é aprender esse relacionamento durante o treinamento. Após o treinamento o modelo é avaliado. Nessa etapa, o modelo faz previsões em um conjunto de testes em que ele tenha acesso apenas as *features*. Uma vez que as respostas para o conjunto de testes já são conhecidas, é possível comparar as previsões realizadas com o valor real para fazer a avaliação da precisão do modelo.

Desse modo, ao treinar o modelo, os dados foram divididos aleatoriamente em conjuntos de treinamento e testes, para obter uma representação de todos os pontos de dados. Assim, os dados foram empiricamente separados em conjunto de treinamento e teste em 80% e 20% respectivamente. Após o treinamento, o modelo criado foi capaz de classificar as *features* com 99% de precisão.

A implementação da *Random Forest* foi realizada utilizando Skicit-learn¹. O Scikit-learn é uma biblioteca de aprendizado de máquina de software livre para a linguagem de programação Python (Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, 2011). Ela apresenta diversos algoritmos de classificação, regressão e *clustering*, incluindo *Random Forest*, *Support Vector Machines*, *k-means*, entre outros, e foi projetado para ser utilizado juntamente com as bibliotecas numéricas, NumPy, e científicas, SciPy, do Python.

Uma das *features* dessa implementação é a possibilidade de visualizar as árvores implementadas. A Figura 7.1 apresenta um exemplo de uma árvore criada para ser utilizada pelo *Random Forest*.

Figura 7.1 – Exemplo de uma árvore retirada da floresta

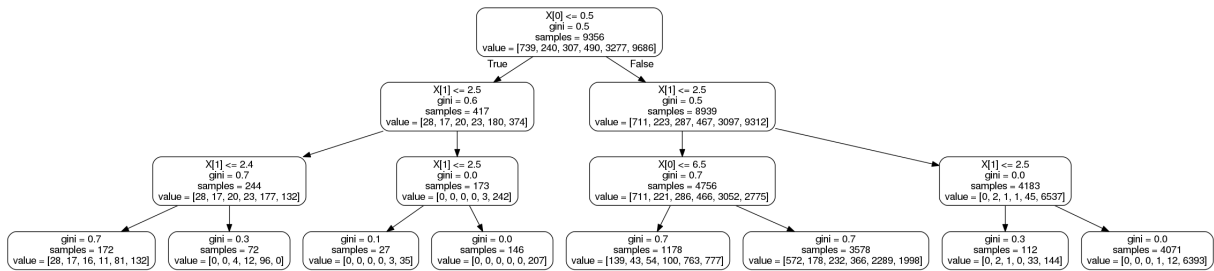


A profundidade dessa árvore é maior do que 15 camadas, por isso fica impossível visualizar os detalhes por essa imagem. Por esse motivo, para obter uma imagem de uma árvore menor, a profundidade da árvore foi limitada. Um exemplo da árvore reduzida pode ser visualizada pela Figura 7.2.

Por esta imagem, é possível visualizar a primeira informação na árvore: a variável e o valor definido para realizar a divisão da árvore. Em seguida, a função para medir a qualidade de uma divisão. A importância de uma *feature* é calculada como a redução total, normalizada, do critério trazido por esse recurso, também sendo conhecido como a importância de Gini (SANDRI; ZUCCOLOTTO, 2008). O critério utilizado foi "gini", que verifica a importância da *feature*, já que, quanto mais alto, mais importante é a *feature*. Em seguida, a quantidade de amostras presentes em determinado nó da árvore, e finalmente um conjunto dos valores de classificação.

¹ <http://scikit-learn.org/>

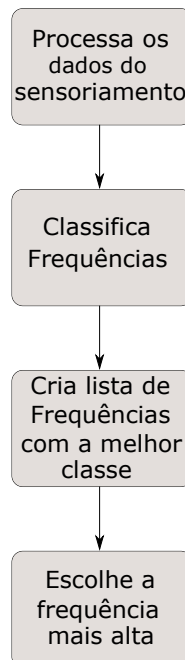
Figura 7.2 – Exemplo de uma árvore reduzida



Além disso, é possível ainda verificar o grau de importância que a *Random Forest* usa para cada *feature*. Um exemplo retirado de uma árvore aleatoriamente é o vetor a seguir.

[0.013269369317574836, 0.1879286823085104, 0.79880194837391483]

Os elementos representam o grau de importância na ordem de cada *feature* utilizada. Assim, eles representam a hora, a frequência, e o ruído base respectivamente. Apesar das primeiras *features*, hora e frequência terem um impacto menor se comparado com o ruído base, elas ainda são de grande importância para a correta tomada de decisão da *Random Forest*.

Figura 7.3 – Fluxograma do *Random Forest*

A Figura 7.3 apresenta as etapas que o *Random Forest* realiza para a escolha da melhor frequência. Primeiramente, os dados recebidos do sensoriamento são analisados e processados, de maneira que o *Random Forest* possa identificar os valores da hora, frequência e ruído base. Em seguida, é realizada a classificação de cada uma das frequências, assim cada frequência

recebe uma classificação de acordo com a tabela 7.4. Após realizada a classificação, as frequências da melhor classe obtida são agrupadas em uma lista. Finalmente, a frequência escolhida é a que se encontra nessa lista de frequências e que possui o valor mais alto.

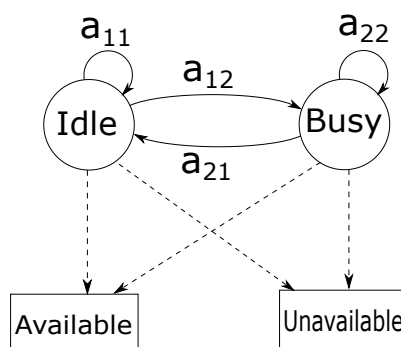
7.1.2 *Hidden Markov Model*

Um *Hidden Markov Model* (HMM) é um modelo estatístico, no qual o sistema que está sendo modelado é considerado um Processo de Markov com estados ocultos, ou estados não observáveis.

A base de dados utilizada é a mesma que foi utilizada pelo *Random Forest* e que foi explicada na seção 7.1. No entanto, os dados brutos foram modificados de maneira diferente para atender as peculiaridades do HMM. Os dados foram modelados de modo a ilustrar uma sequência de observação da ocupação do espectro. A ocupação do espectro foi modelada como uma sequência binária com símbolo 1 (um), denotando que o espectro se encontra ocupado naquele instante em particular, e um símbolo 0 (zero), que indica que o espectro se encontra livre. Os canais foram avaliados considerando o limiar de -107 dBm (GHOSH; ROY; RAO, 2012). O canal é considerado ocioso com base na potência recebida. Se a potência recebida for maior que o limiar de detecção, o canal é considerado ocupado e se a potência recebida for menor que o limiar de detecção, o canal é considerado ocioso.

A implementação do *Hidden Markov Model* foi realizada utilizando o HMMs². O HMMs é uma biblioteca para *Hidden Markov Model* desenvolvida para Python. Essa biblioteca é de fácil utilização, além de ser uma biblioteca de propósito geral que implementa todos os métodos necessários para o treinamento de um modelo HMM.

Figura 7.4 – Modelagem para *Hidden Markov Model*



² <https://github.com/lopatovsky/HMMs>

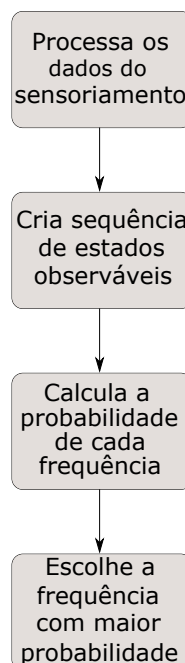
O modelo utilizado é apresentado pela Figura 7.4. Neste modelo, foram definidos dois estados para representar a matriz de transição de estados, os estados em uso e vago, que indicam a situação do canal em determinado momento. Ademais, ainda foram definidos dois estados que representam a matriz de emissão, que indicam se o canal está disponível para uso ou indisponível. Um modelo distinto foi treinado e utilizado para cada frequência em um intervalo de 2,5MHz. Um exemplo de parâmetros treinados em um modelo de uma faixa específica de frequência pode ser visualizado pela Tabela 7.5.

Tabela 7.5 – Exemplo de parâmetros de um Modelo

Parâmetros	Valor
π	$(0,198868 \quad 0,801132)$
A	$\begin{pmatrix} 0,642655 & 0,357345 \\ 0,588695 & 0,411305 \end{pmatrix}$
B	$\begin{pmatrix} 0,163757 & 0,836243 \\ 0,298862 & 0,701138 \end{pmatrix}$

Após a definição do modelo, foi utilizado os dados modelados como estados observáveis para utilizar o algoritmo de Baum Welch, com a finalidade de encontrar os parâmetros do modelo. Com os parâmetros encontrados, foi utilizado o algoritmo de Viterbi para gerar a provável sequência de estados a partir do modelo. Assim, a frequência escolhida será a frequência que obtiver a maior probabilidade de estar disponível.

Figura 7.5 – Fluxograma do *Hidden Markov Model*



A Figura 7.5 apresenta as etapas que o *Hidden Markov Model* realiza para a escolha da melhor frequência. Primeiramente, os dados recebidos do sensoriamento são analisados e processados, de maneira que o *Hidden Markov Model* possa identificar os valores da hora, frequência e ruído base. Em seguida, é criada a sequência de estados observáveis baseados no ruído base de cada frequência. Nesta etapa, é utilizado um valor de *threshold* pré-determinado, se o valor do ruído base for maior que esse limite, a sequência de observação receberá o valor um, caso contrário o valor será zero. Com a sequência de estados observáveis determinada, é calculado a probabilidade do estado de emissão ser disponível para cada uma das frequências. Finalmente, a frequência escolhida é a que possui maior probabilidade do estado de emissão ser disponível.

7.1.3 Rede Neural Artificial

A modelagem da RNA foi feita por Marques (2016). O treinamento utilizou uma base onde os dados foram coletados durante uma semana. Nesses 7 dias, os dados coletados do espectro local continham informações de hora coletada, frequência e ruído base. As faixas do espectro que foram analisadas estão entre 800MHz a 6GHz. Esses dados foram particionados em 4/7 que foram utilizados para o treinamento e 3/7 para a validação, resultando em aproximadamente 57% para treinamento e 43% para validação.

Diversos testes foram realizados a fim de determinar os parâmetros adequados para a definição da RNA. A tabela 7.6 apresenta os parâmetros escolhidos para a RNA criada.

Tabela 7.6 – Configuração da RNA Marques (2016)

Variáveis	Valores
Arquitetura	<i>Multilayer perceptron</i> (MLP)
Função de Ativação	<i>Sigmoid</i>
Treinamento	Supervisionado
Processo de Aprendizagem	Época
Neurônios Intermediários	34
Neurônios de saída	1
Camadas escondidas	2
<i>Momentum</i>	0,0
Taxa de aprendizado	0,7
Erro médio Quadrático	0,0004

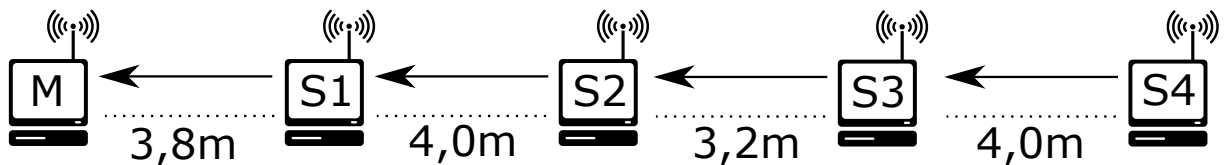
Marques (2016) enfatiza que após o treinamento realizado com esses parâmetros, a RNA foi capaz de determinar a melhor frequência em determinado momento.

7.2 Topologias

Para a realização dos testes, foram definidas três topologias. Essas topologias indicam o posicionamento dos nós no ambiente de testes e influenciam diretamente no resultado dos testes. Para garantir que a comunicação entre os nós fosse realizada somente entre os nós definidos em cada topologia, a potência de transmissão e a sensibilidade do rádio dos nós foi ajustada de acordo com a necessidade.

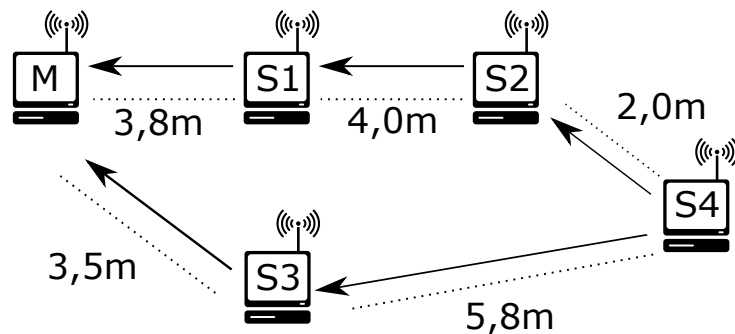
A primeira é uma topologia linear que possui 5 nós, um nó *Master* e quatro nós *Slaves*. Essa topologia pode ser visualizada pela Figura 7.6 e foi utilizada para determinar a qualidade da transmissão com diversos saltos. O espaçamento entre os nós variou entre 3 a 4 metros. A comunicação dessa topologia é realizada a partir do nó S4 até o nó M, desta forma, os pacotes são encaminhados através de toda a rede passando por todos os nós.

Figura 7.6 – Topologia em linha



A segunda topologia possui duas rotas possíveis para o encaminhamento dos pacotes. Essa topologia conta com cinco nós, um nó *Master* e quatro nós *Slaves* e pode ser visualizada pela Figura 7.7. Essa topologia foi utilizada para determinar a qualidade da transmissão com mais de uma rota possível. Nessa topologia, o nó S4 pode fazer o encaminhamento de pacotes utilizando a rota passando pelos nós S2 e S1, chegando ao nó M, ou encaminhar para o nó S3 que encaminhará diretamente para o nó M.

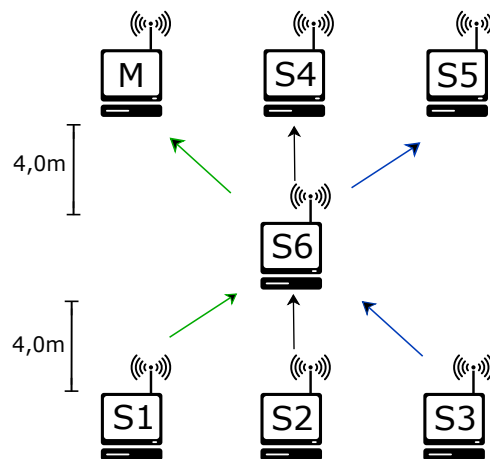
Figura 7.7 – Topologia com duas rotas possíveis



A terceira é uma topologia com um nó intermediário que realiza o encaminhamento dos pacotes de toda a comunicação da rede. Essa topologia consiste de sete nós, sendo um nó *Master*

e seis nós Slaves. Essa topologia pode ser visualizada pela da Figura 7.8 e foi utilizada para determinar a robustez da transmissão, com cargas de transmissão concentradas em um ponto da rede. O nó S6 é capaz de realizar o encaminhamento para qualquer nó da rede. A comunicação da rede é realizada do nó S1 para o M, do nó S2 para o nó S4 e do nó S3 para o nó S5, com o nó S6 realizando o encaminhamento de todos os pacotes. A distância do nó S6 para qualquer outro nó da rede é de aproximadamente quatro metros.

Figura 7.8 – Topologia com um nó intermediário para encaminhamento



Geração de tráfego

Para avaliar o comportamento do *framework* em diferentes tráfegos de mensagens, foram definidos três diferentes padrões de geração de tráfego para os testes realizados. O primeiro padrão de geração de tráfego utiliza um intervalo constante de 50 milissegundos para a geração dos pacotes. O segundo padrão de geração de tráfego utiliza um intervalo constante de 12 milissegundos para a geração dos pacotes. Por fim, o terceiro padrão de geração de tráfego utiliza um intervalo constante de 8 milissegundos para a geração dos pacotes. A tabela 7.7 apresenta os padrão de geração de tráfego baixo, médio e alto e a geração de aproximadamente 20, 83, 125 pacotes por segundo respectivamente.

Tabela 7.7 – Padrões de geração de tráfego

Intervalo de geração de mensagens	Tipo de tráfego	Pacotes por segundo
50 ms	Baixo - L	20
12 ms	Médio - M	83
8 ms	Alto - H	125

7.3 Métricas avaliadas

Diferentes métricas foram determinadas com a finalidade de realizar a avaliação da solução proposta. As três métricas definidas foram: Taxa de Entrega, Latência e *Handoff*.

1. *Taxa de Entrega*: métrica que possui a finalidade de mensurar a relação da quantidade de pacotes enviados pelo nó origem pela quantidade de pacotes recebidos pelo nó de destino.
2. *Latência*: métrica que permite mensurar o tempo para que uma sequência de *bits* seja transmitida entre dois nós. Essa métrica foi escolhida para verificar o tempo utilizado para a transmissão de uma mensagem.
3. *Handoff*: é o processo de transferência de uma sessão de dados ativa de uma frequência de operação para outra frequência de operação. Essa métrica foi utilizada a fim de determinar a quantidade de vezes que houve alteração da frequência de operação.

7.4 Resultados

Os cenários de testes utilizam os modelos de aprendizado de máquina para decisão do melhor canal, protocolos da camada de rede, topologias da rede físicas e padrões de geração de tráfego. Nos cenários são avaliadas as métricas de latência, taxa de entrega e *Handoff*. Os equipamentos utilizados para os testes são compostos de USRP, modelos B200 e B210, conectadas em computadores de uso geral. Os experimentos foram realizados em ambientes fechados, e por isso propensos a sofrer maior interferência do que em ambientes abertos.

7.4.1 *Handoff*

Para a avaliação do *Handoff*, os três métodos de aprendizado de máquina foram utilizados, sendo eles RnF, HMM e RNA. Do mesmo modo que os padrões de tráfego utilizados foram H para o tipo alto, M para o tipo médio e L para o tipo baixo. Os gráficos foram divididos de acordo com os padrões de tráfego utilizados, mostrando o resultado da taxa de entrega para cada método de aprendizado de máquina empregado. Cada gráfico mostra a frequência escolhida por cada um dos métodos de aprendizado de máquina, nas trinta repetições realizadas para cada teste.

Para uma melhor análise do *Handoff*, foram definidas classes para agrupar as frequências analisadas, estas classes foram apenas criadas para uma avaliação sintetizada dos resultados obtidos. A classe *C1* corresponde às frequências mais baixas do espectro no intervalo de 0,8GHz a 5,8GHz, a classe *C7* por sua vez corresponde às frequências mais altas. As frequências foram definidas em ordem crescente de acordo com as classes. A definição completa das classes pode ser visualizada pela Tabela 7.8.

Tabela 7.8 – Definição das classes para a métrica de *Handoff*

Intervalo de Frequências (MHz)	Classe
800 – 1499	C1
1500 – 2199	C2
2200 – 2999	C3
3000 – 3699	C4
3700 – 4399	C5
4400 – 5199	C6
5200 – 5800	C7
6000	CCC

Os resultados obtidos pelas topologias 1, 2 e 3 são apresentados pelas Figuras 7.9, 7.10 e 7.11 respectivamente.

Figura 7.9 – *Handoff* topologia 1 com AODV

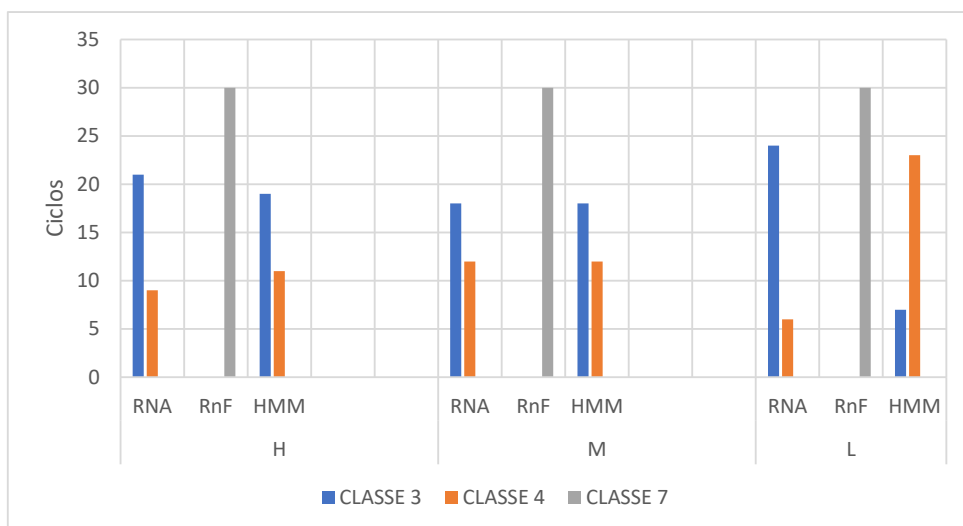
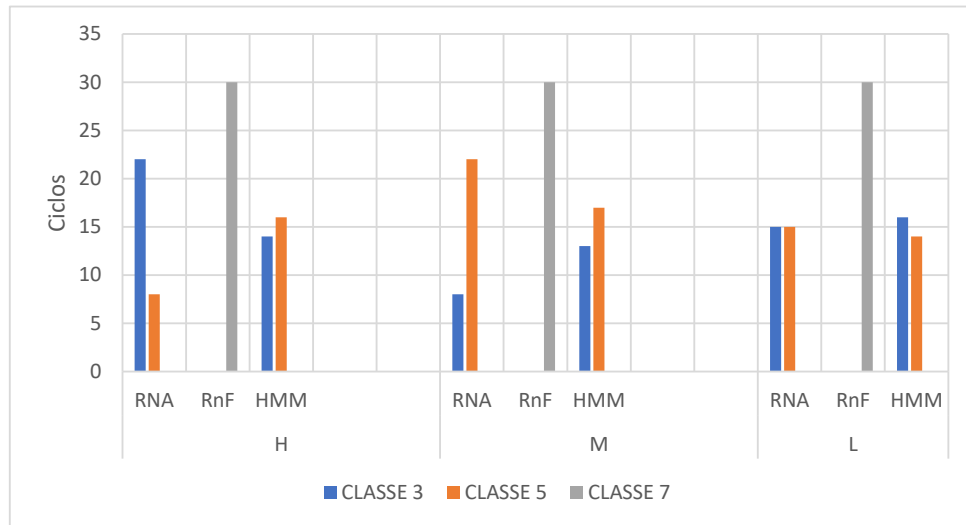
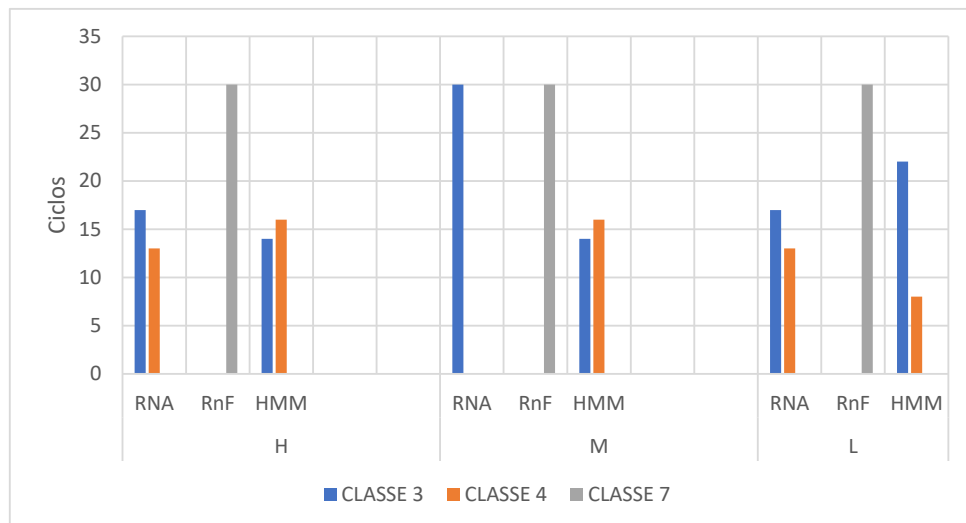


Figura 7.10 – *Handoff* topologia 2 com AODVFigura 7.11 – *Handoff* topologia 3 com AODV

Não houve uma grande variação das frequências escolhidas em cada método individualmente, somente houve uma maior variação da frequência entre os diferentes métodos de aprendizado de máquina.

É possível perceber pelos gráficos que os métodos HMM e RNA escolheram as frequências nas proximidades de $3GHz$ na maior partes das execuções dos testes. Enquanto que o método RnF manteve-se em frequências mais altas, prevalecendo na maioria dos casos a frequência de $5,8GHz$. Embora em alguns casos as frequências escolhidas pelos métodos HMM e RnF foram classificadas em diferentes classes de acordo com a classificação proposta, a variação ocorreu dentro do limite das classes. Em alguns casos variando de $2,8GHz$ a $3,1GHz$. Por esse motivo as classes utilizadas variaram principalmente entre a classe 3 e classe 4. De certo modo,

quando se observa os resultados de uma visão geral, os métodos HMM e RNA foram muito próximos nas escolhas das frequências de transmissão.

7.4.2 Latência

Os resultados dessa seção analisam a latência média do *framework*. Os três métodos de aprendizado de máquina foram utilizados, RnF, HMM e RNA. Foram executadas trinta repetições e de cada uma delas foi calculado o tempo médio de latência dos nós. Posteriormente, foi calculada a média desses resultados para chegar ao valor apresentado nos gráficos. Os padrões de geração de tráfego utilizados foram H para o tipo alto, M para o tipo médio e L para o tipo baixo. Os gráficos foram divididos de acordo com os padrões de tráfego utilizados, mostrando o resultado da latência para cada método de aprendizado de máquina empregado.

Na topologia 1 os resultados são apresentados pela Figura 7.12 onde foi utilizado AODV e pela Figura 7.13 onde foi utilizado o OLSR.

Figura 7.12 – Latência topologia 1 com AODV

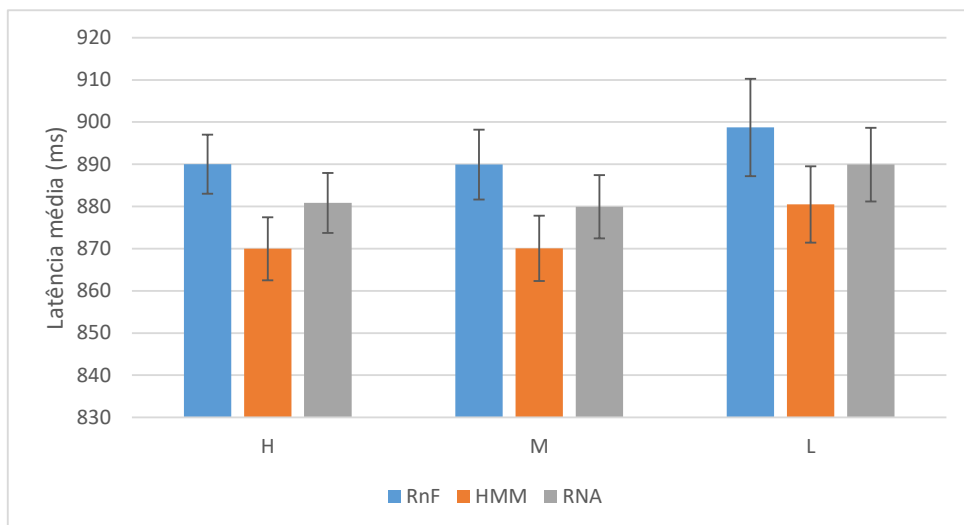
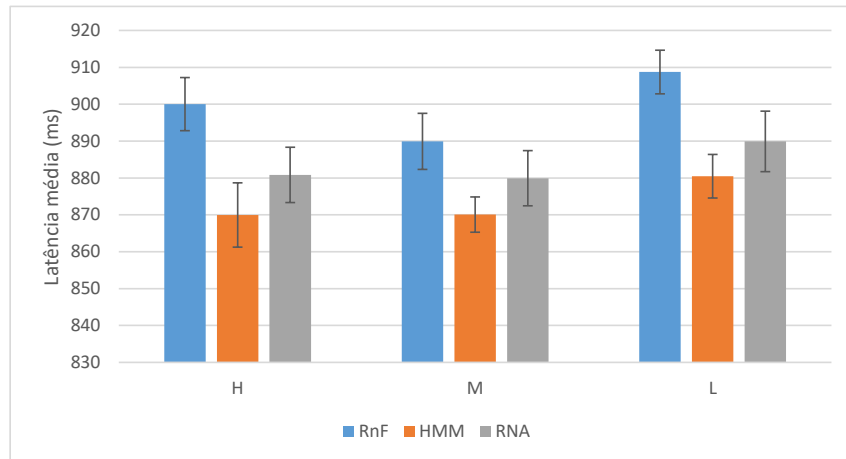


Figura 7.13 – Latência topologia 1 com OLSR



Na topologia 2 os resultados são apresentados pela Figura 7.14 onde foi utilizado AODV e pela Figura 7.15 onde foi utilizado o OLSR.

Figura 7.14 – Latência topologia 2 com AODV

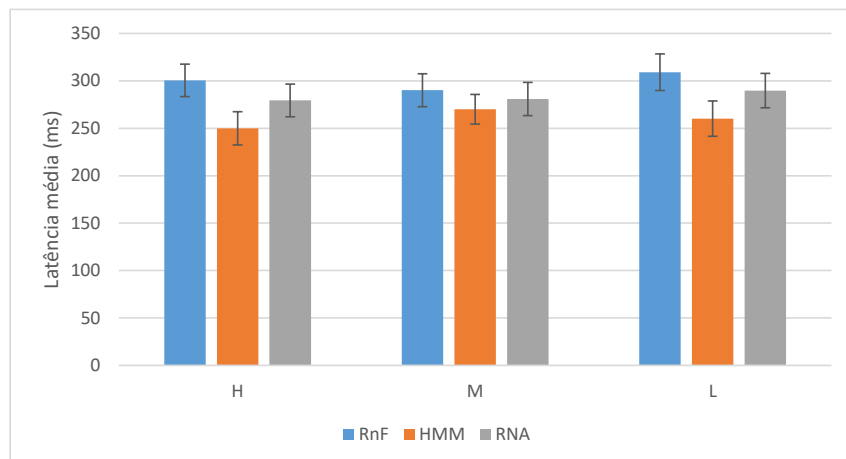
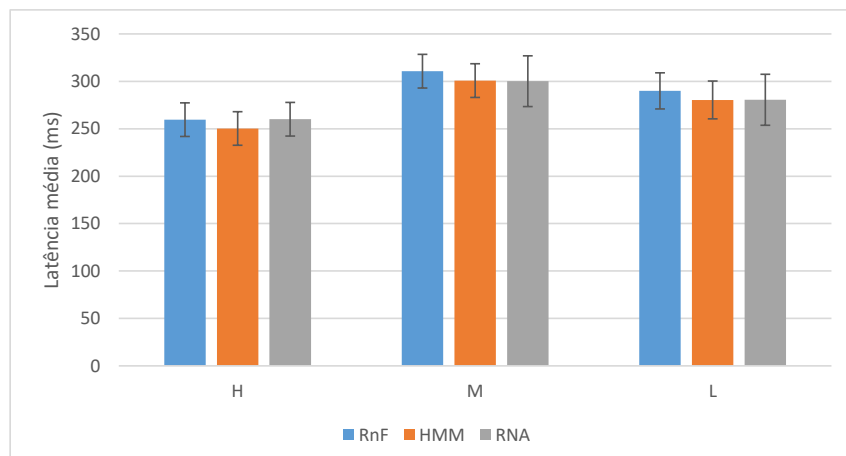


Figura 7.15 – Latência topologia 2 com OLSR



Por fim, para a topologia 3 os resultados são apresentados pela Figura 7.16 onde foi utilizado AODV e pela Figura 7.17 onde foi utilizado o OLSR.

Figura 7.16 – Latência topologia 3 com AODV

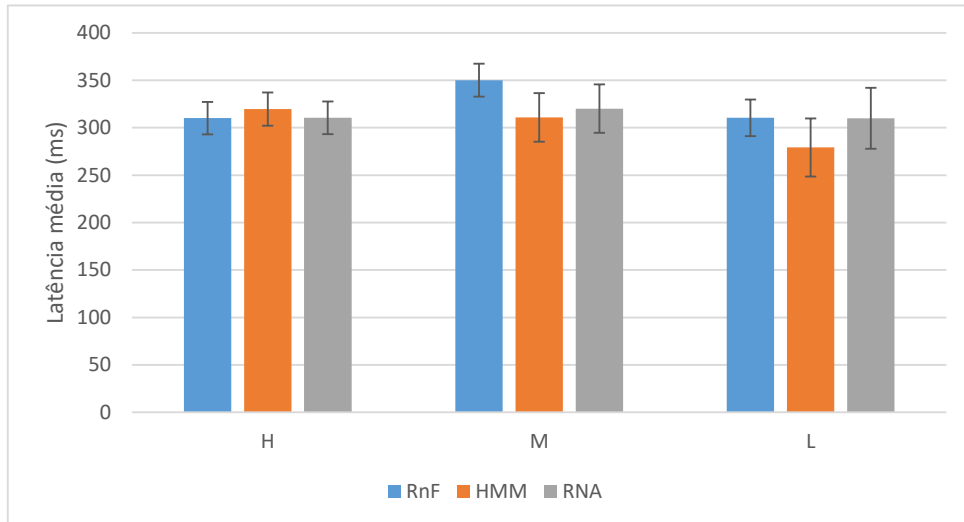
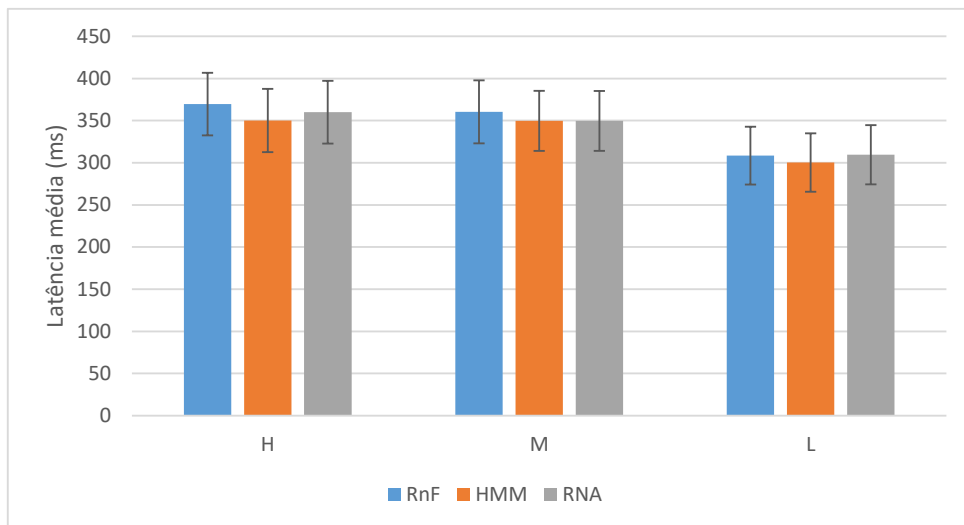


Figura 7.17 – Latência topologia 3 com OLSR



Observa-se que o método de aprendizado de máquina que proporcionou menores latências foi na maioria das vezes o HMM. Contudo, a latência verificada pelos outros métodos é bastante similar ao HMM. Isso acontece pois todos os métodos fazem uso do mesmo meio de transmissão, e também pela natureza da implementação do *framework*. A diferença pequena apresentada está relacionada com as frequências determinadas por cada método. Por esse motivo, na maior parte dos casos, o HMM se destaca como o melhor método em relação aos outros métodos analisados. O protocolo MAC utilizado não foi completamente otimizado de maneira que não existe reserva de canal e por esse motivo a latência tende a possuir valores altos.

7.4.3 Taxa de Entrega

A taxa de entrega é analisada pelos resultados apresentados nessa seção, onde os três métodos de aprendizado de máquina foram utilizados, *Random Forest*, *Hidden Markov Model* e Redes Neurais Artificiais. Os padrões de geração de tráfego utilizados foram H para o tipo alto, M para o tipo médio e L para o tipo baixo. Os gráficos foram divididos de acordo com os padrões de geração de tráfego, mostrando o resultado da taxa de entrega para cada método de aprendizado de máquina empregado. O total de trinta repetições foram executadas e em cada uma delas foi calculado o tempo médio da taxa de entrega. Ao final das repetições, foi calculada a média desses trinta valores para chegar ao valor apresentado nos gráficos.

A Figura 7.18 apresenta os resultados da topologia 1 utilizando AODV e a Figura 7.19 os resultados utilizando o OLSR.

Figura 7.18 – Taxa de entrega topologia 1 com AODV

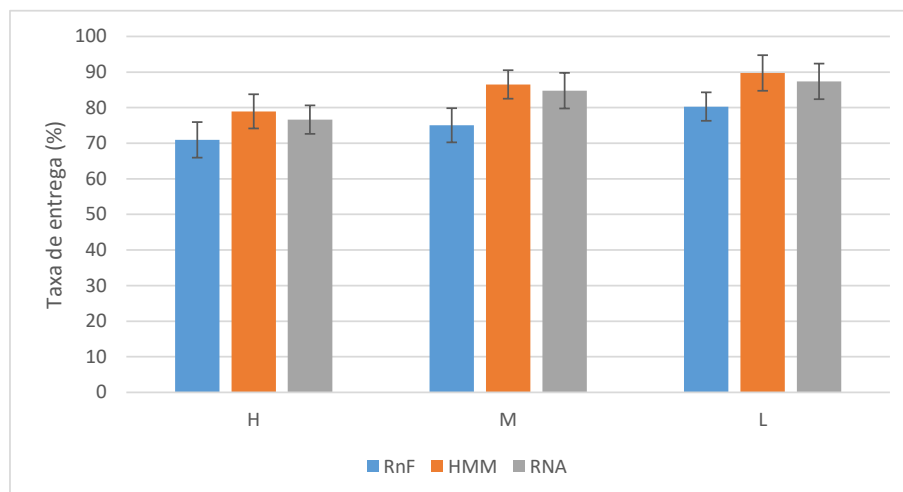
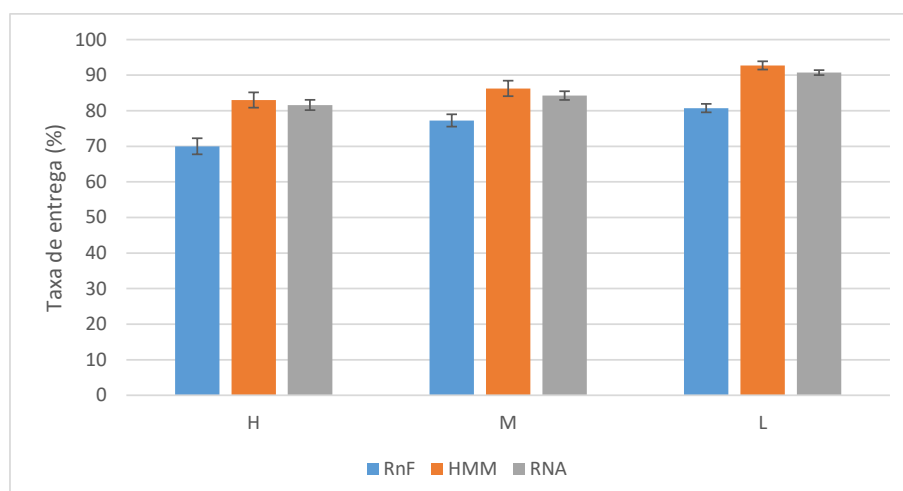


Figura 7.19 – Taxa de entrega topologia 1 com OLSR



Na topologia 2 os resultados são apresentados pela Figura 7.20 onde foi utilizado AODV e pela Figura 7.21 onde foi utilizado o OLSR.

Figura 7.20 – Taxa de entrega topologia 2 com AODV

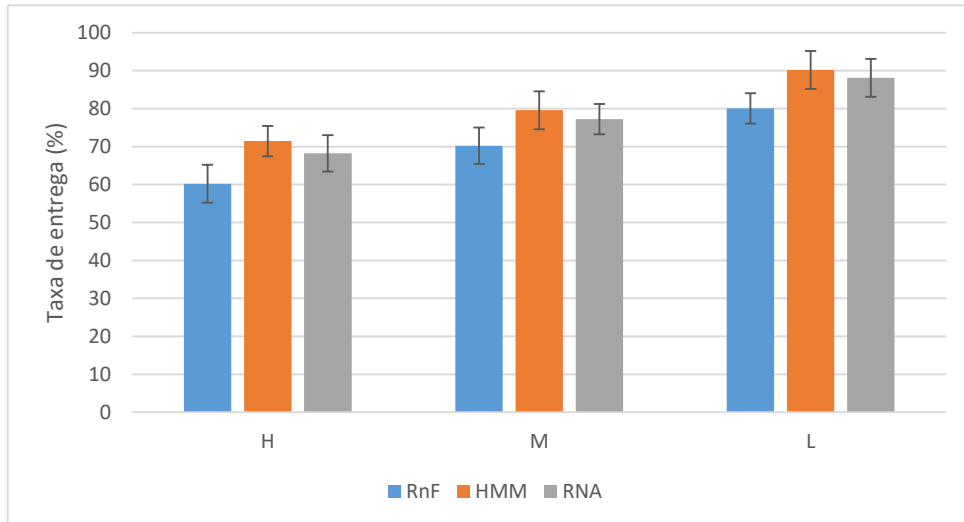
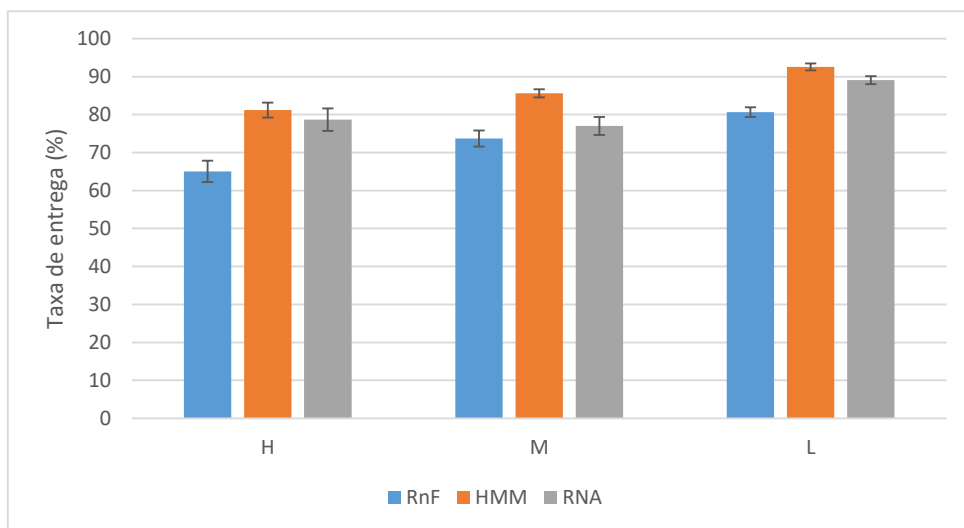


Figura 7.21 – Taxa de entrega topologia 2 com OLSR



Por último, para a topologia 3 os resultados são apresentados pela Figura 7.22 onde foi utilizado AODV e pela Figura 7.23 onde foi utilizado o OLSR.

Figura 7.22 – Taxa de entrega topologia 3 com AODV

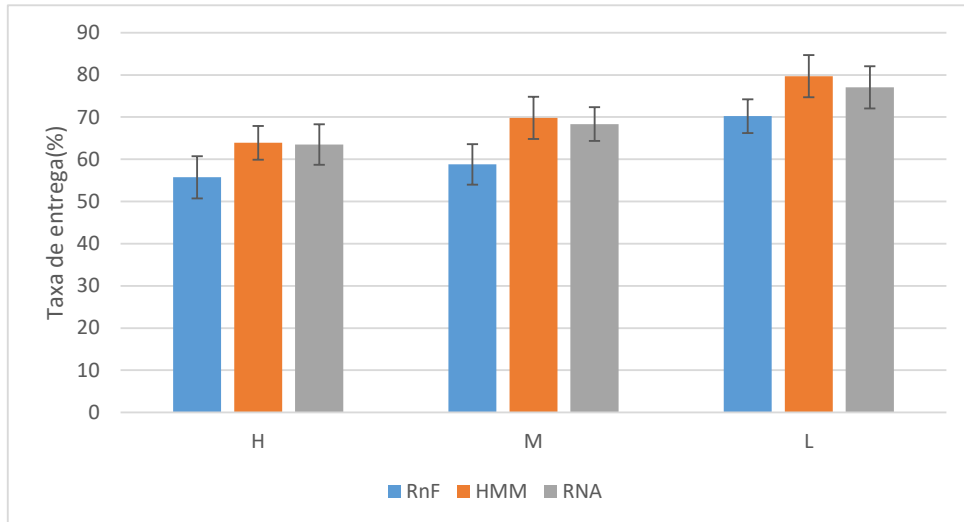
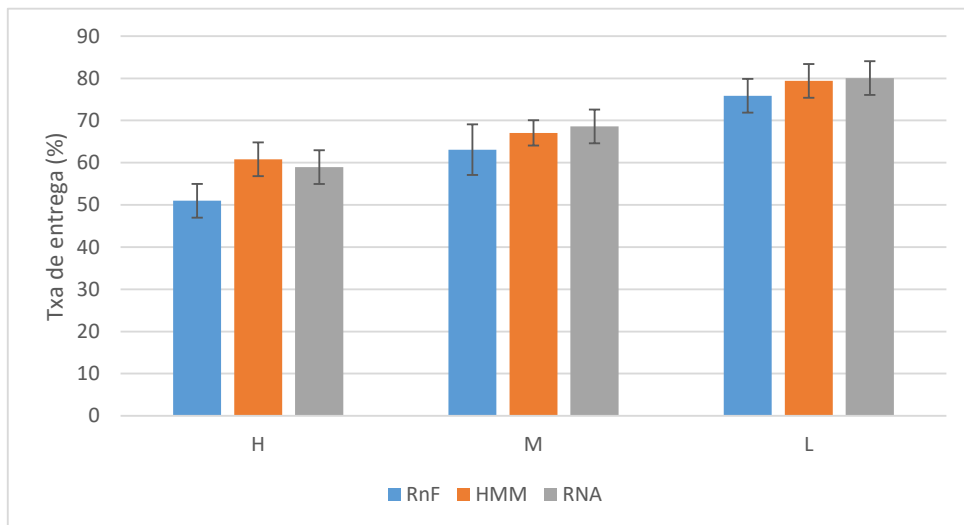


Figura 7.23 – Taxa de entrega topologia 3 com OLSR



O OLSR é um bom algoritmo para redes densas em que existem muitas trocas de mensagens entre nós no mesmo alcance. Não houveram diferenças significativas nos resultados dos protocolos AODV e OLSR. Isso teve ocorrência por causa dos experimentos terem sido realizados em ambientes reais. A quantidade de recursos disponíveis para os testes não foram suficientes para oferecer grandes quantidades de nós para a criação de redes densas. Por este motivo, os resultados do AODV e OLSR foram bem parecidos. Em todos os testes realizados existe uma menor taxa de entrega para o padrão de tráfego alto (H). Isso ocorre pois existe maior concorrência pelo uso do canal com intervalos menores entre o envio das mensagens.

Observa-se que o método de aprendizado de máquina que proporcionou maiores taxas de entrega foi na maioria das vezes o HMM. Contudo, a taxa de entrega verificada pela RNA é bastante similar ao HMM. Isso ocorre pois ambos os métodos definiram faixas do espectro bem próximas, sendo que as vezes coincidiram em utilizar a mesma faixa para a transmissão dos dados. O método do RnF por sua vez, manteve-se em frequências mais altas. Como existe uma maior tendência em ambientes internos de existir interferências, essas faixas mais elevadas do espectro podem ter ocasionado uma menor taxa de entrega, comparado aos outros modelos. Ainda é preciso levar em consideração a recente popularização das faixas de frequências de $5,8\text{GHz}$ principalmente para o uso do Wi-Fi doméstico. Tendo em vista que o mapeamento que foi utilizado para o treinamento de todos os métodos de aprendizado de máquina foi previamente obtido, possivelmente ocorreram alterações nos padrões da rede, que podem não ter sido levadas em consideração afetando assim o desempenho do RnF.

8 CONCLUSÃO

Devido a grande utilização da faixa ISM e o mau uso do espectro, este trabalho apresentou um *framework*, que é uma ampliação de uma arquitetura para Redes de Rádios Cognitivos. Este *framework* possibilita a criação de Redes de Rádios Cognitivos capazes de executar corretamente as etapas de sensoriamento do espectro, decisão do espectro, compartilhamento do espectro e mobilidade do espectro.

O *framework* implementou dois novos métodos para a decisão do espectro baseados em aprendizado de máquina. O primeiro método foi o *Random Forest* e o segundo método foi o *Hidden Markov Model*. Esses métodos foram capazes de determinar as faixas de frequência mais adequadas para o melhor desempenho de transmissão. Com a influência das recentes pesquisas, visando a criação de abordagens que incluem a utilização dos Rádios Cognitivos em redes multi-hop, o *framework* utilizou dois protocolos de roteamento, AODV e OLSR, para que este pudesse suportar a comunicação multi-hop.

Três diferentes métodos de decisão do espectro foram avaliados em uma rede multi-hop. Os métodos avaliados foram o *Random Forest*, *Hidden Markov Model* e Rede Neural Artificial. As topologias utilizadas para os testes verificam o comportamento dos métodos de decisão e dos protocolos de roteamento em diferentes situações de transmissão de dados, uma topologia linear, uma topologia com dois caminhos possíveis e uma topologia onde um nó teve uma grande carga de transmissões.

Os resultados mostraram que o método de decisão do espectro baseado no *Hidden Markov Model* obteve a melhor taxa de entrega média nos cenários avaliados, ao custo de uma latência média semelhante aos outros métodos. Devido a natureza dos protocolos de roteamento em serem focados para redes com topologias mais complexas, não houve uma diferença notória desses protocolos nos cenários analisados, uma vez que as topologias utilizadas contaram com uma quantidade não muito grande de nós, devido a quantidade de recursos disponíveis.

Como propostas para os trabalhos futuros, as atividades a seguir foram definidas para preencher as lacunas identificadas atualmente neste *framework*.

- Pretende-se avaliar cenários em ambientes externos, uma vez que existe uma diferença em relação a interferência gerada pelos próprios nós entre ambientes internos e externos.
- Avaliar topologias mais complexas. Topologias que de preferência possuam uma maior quantidade de nós na Rede de Rádios Cognitivos, para ser possível criar redes densas.

- Realizar uma nova avaliação do ambiente local, para que os algoritmos de aprendizado de máquina sejam treinados com o padrão atual do espectro.
- Definir novos métodos de aprendizado de máquina, como por exemplo o *concept drift*, que utilizem treinamento semi-supervisionado ou não supervisionado, dessa maneira a avaliação do ambiente local poderá não causar tanto impacto no resultado final do *framework*.
- Implementar interfaces de redes virtuais como por exemplo TUN/TAP.
- Realizar a implementação da camada de transporte.

REFERÊNCIAS

- AKYILDIZ, I.; LEE, W.-Y.; CHOWDHURY, K. Spectrum management in cognitive radio ad hoc networks. **IEEE Network**, v. 23, n. 4, p. 6–12, 2009. ISSN 0890-8044. Disponível em: <<http://ieeexplore.ieee.org/document/5191140/>>.
- AKYILDIZ, I. F. et al. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. **Computer Networks**, Elsevier, v. 50, n. 13, p. 2127–2159, sep 2006. ISSN 13891286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1389128606001009?via=IiD>>.
- AKYILDIZ, I. F. et al. A Survey on Spectrum Management in Cognitive Radio Networks. **Communications Magazine, IEEE**, v. 46, n. 4, p. 40–48, apr 2008. ISSN 0163-6804. Disponível em: <<http://ieeexplore.ieee.org/document/4481339/http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=4481>>.
- BANTOUNA, A. et al. An overview of learning mechanisms for cognitive systems. **EURASIP Journal on Wireless Communications and Networking**, Springer International Publishing, v. 2012, n. 1, p. 22, 2012. ISSN 1687-1499. Disponível em: <<http://jwcn.eurasipjournals.com/content/2012/1/22>>.
- BAUM, L. E. et al. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. **The Annals of Mathematical Statistics**, v. 41, n. 1, p. 164–171, 1970. ISSN 0003-4851. Disponível em: <<http://www.jstor.org/stable/2239727http://projecteuclid.org/euclid.aoms/1177697196>>.
- BAYES, T.; PRICE, R. An Essay towards Solving a Problem in the Doctrine of Chances By the Late Rev. Mr. Bayes. **Philosophical Transactions (1683-1775)**, Springer India, v. 53, n. 4, p. 370–418, apr 1763. ISSN 02607085. Disponível em: <<http://link.springer.com/10.1007/BF02883540>>.
- BILMES, J. **A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models**. [S.l.], 1998.
- BKASSINY, M.; LI, Y.; JAYAWEERA, S. K. A survey on machine-learning techniques in cognitive radios. **IEEE Communications Surveys and Tutorials**, v. 15, n. 3, p. 1136–1159, 2013. ISSN 1553877X. Disponível em: <<http://ieeexplore.ieee.org/document/6336689/>>.
- BLOSSOM, E. **GNU Radio The Free & Open Software Radio Ecosystem**. 2016. Disponível em: <<http://gnuradio.org/>>.
- BREIMAN, L. Bagging Predictors. **Machine Learning**, Kluwer Academic Publishers, v. 24, n. 421, p. 123–140, aug 1996. ISSN 0885-6125. Disponível em: <<http://link.springer.com/10.1007/BF00058655>>.
- BREIMAN, L. Random forests. **Machine Learning**, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 08856125. Disponível em: <<http://link.springer.com/10.1023/A:1010933404324>>.
- CABRIC, D.; MISHRA, S.; BRODERSEN, R. Implementation issues in spectrum sensing for cognitive radios. **Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers**, IEEE, v. 1, p. 772–776, 2004. ISSN 10586393. Disponível em: <<http://ieeexplore.ieee.org/document/1399240/>>.

- CACCIAPUOTI, A. S.; AKYILDIZ, I. F.; PAURA, L. Optimal primary-user mobility aware spectrum sensing design for cognitive radio networks. **IEEE Journal on Selected Areas in Communications**, v. 31, n. 11, p. 2161–2172, nov 2013. ISSN 07338716. Disponível em: <<http://ieeexplore.ieee.org/document/6518484/>>.
- CESANA, M.; CUOMO, F.; EKICI, E. Routing in cognitive radio networks: Challenges and solutions. **Ad Hoc Networks**, v. 9, n. 3, p. 228–248, 2011. ISSN 15708705. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S157087051000079X>>.
- CHEN, Y. S.; HONG, J. S. A relay-assisted protocol for spectrum mobility and handover in cognitive LTE networks. **IEEE Systems Journal**, v. 7, n. 1, p. 77–91, mar 2013. ISSN 19328184. Disponível em: <<http://ieeexplore.ieee.org/document/6317127/>>.
- CHEN, Z.; QIU, R. C. Prediction of channel state for cognitive radio using higher-order hidden Markov model. **Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)**, IEEE, p. 276–282, mar 2010. ISSN 07347502. Disponível em: <<http://ieeexplore.ieee.org/document/5453870/http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5453870>>.
- 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, jan 2008. ISSN 07338716. Disponível em: <<http://ieeexplore.ieee.org/document/4413149/>>.
- CHRISTODOULOU, C. G. Reconfigurable antennas in cognitive radio that can think for themselves? In: **Proceedings - 2009 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, MAPE 2009**. IEEE, 2009. p. k-1–k-3. ISBN 9781424440740. Disponível em: <<http://ieeexplore.ieee.org/document/5355912/>>.
- CLAUSEN, T.; JACQUET, P. **Optimized Link State Routing Protocol (OLSR)**. IETF, 2003. RFC 3626 (Experimental). (Request for Comments, 3626). Disponível em: <<http://www.ietf.org/rfc/rfc3626.txt>>.
- CORMIO, C.; CHOWDHURY, K. R. A survey on MAC protocols for cognitive radio networks. **Ad Hoc Networks**, Elsevier, v. 7, n. 7, p. 1315–1329, sep 2009. ISSN 15708705. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S1570870509000043>>.
- CORREIA, L. H. A. et al. A framework for cognitive radio wireless sensor networks. **IEEE Symposium on Computers and Communications (ISCC)**, IEEE, p. 611–616, jul 2012. ISSN 15301346. Disponível em: <<http://ieeexplore.ieee.org/document/6249364/http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6249364>>.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Mathematics of Control, Signals, and Systems**, Springer-Verlag, v. 2, n. 4, p. 303–314, dec 1989. ISSN 0932-4194. Disponível em: <<http://link.springer.com/10.1007/BF02551274>>.
- CYPRIANI, M. et al. Open wireless positioning system: A Wi-Fi-based indoor positioning system. In: **IEEE Vehicular Technology Conference**. IEEE, 2009. p. 1–5. ISBN 9781424425150. ISSN 15502252. Disponível em: <<http://ieeexplore.ieee.org/document/5378966/>>.

DAVENPORT, W. B. **Probability and Random Processes**. [S.l.]: Oxford University Press, 1970. 46–49 p. ISBN 0198572220.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. **Journal of the Royal Statistical Society Series B (Methodological)**, v. 39, n. 1, p. 1–38, 1977. ISSN 00359246. Disponível em: <<http://www.jstor.org/stable/2984875>><http://www.jstor.org/stable/2984875>{\%}5Cn<http://www.jstor.org/page/info/about/policies/terms.jsp>{\%}5Cn<http://www.jstor.org>.

DILLINGER, M.; MADANI, K.; ALONISTIOTI, N. **Software defined radio : architectures, systems, and functions**. [S.l.]: Wiley, 2003. 416 p. ISBN 9780470851647.

DROZD, A. L. et al. Computational electromagnetics applied to analyzing the efficient utilization of the RF transmission hyperspace. In: **2005 IEEE/ACES International Conference on Wireless Communications and Applied Computational Electromagnetics**. IEEE, 2005. v. 2005, p. 1080–1088. ISBN 0780390687. ISSN 10569170. Disponível em: <<http://ieeexplore.ieee.org/document/1469769/>>.

DUDLEY, S. M. et al. **Practical issues for spectrum management with cognitive radios**. 2014. 242–264 p. Disponível em: <<http://ieeexplore.ieee.org/document/6728604/>>.

ETTUS, M. **Ettus Research**. USA: [s.n.], 2017. National Instruments. Acessado em: 05-05-2017. Disponível em: <<http://www.ettus.com/site/about>>.

FCC. **Norm, FCC 03-322**. Califórnia: Federal Communications Commission, 2003. Disponível em: <https://apps.fcc.gov/edocs{_}public/attachmatch/FCC-03-322A1.>

FLAMMINI, A. et al. Wired and wireless sensor networks for industrial applications. **Microelectronics Journal**, v. 40, n. 9, p. 1322–1336, 2009. ISSN 00262692.

FORNEY, G. D. The Viterbi Algorithm. **Proceedings of the IEEE**, v. 61, n. 3, p. 268–278, 1973. ISSN 15582256. Disponível em: <<http://ieeexplore.ieee.org/document/1450960/>>.

FREUND, Y.; SCHAPIRE, R. R. E. Experiments with a New Boosting Algorithm. **International Conference on Machine Learning**, p. 148–156, 1996. ISSN 0706-652X, 1205-7533. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.6252>>.

FUTEBOL, P. **FUTEBOL**. Brasil: [s.n.], 2018. National Instruments. Acessado em: 05-05-2018. Disponível em: <<http://www.ict-futebol.org.br/>>.

GHOSH, C.; ROY, S.; RAO, M. B. Modeling and validation of channel idleness and spectrum availability for cognitive networks. **IEEE Journal on Selected Areas in Communications**, v. 30, n. 10, p. 2029–2039, nov 2012. ISSN 07338716. Disponível em: <<http://ieeexplore.ieee.org/document/6331692/>>.

GRIESMER, J. H.; MCCARTHY, J. Arthur Samuel: Pioneer in machine learning. **IBM Journal of Research and Development**, IBM Corp., v. 36, n. 3, p. 328–328, may 1992. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5389711>>.

HORNE, W. D. Adaptive spectrum access: using the full spectrum space. In: **Proc. Telecommunications Policy Research Conference (TPRC)**. [S.l.: s.n.], 2003.

HUANG, Y. et al. Design of learning engine based on support vector machine in cognitive radio. In: **Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009**. IEEE, 2009. p. 1–4. ISBN 9781424445073. Disponível em: <<http://ieeexplore.ieee.org/document/5363127/>>.

ILERI, O. et al. Demand responsive pricing and competitive spectrum allocation via a spectrum server. In: **2005 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005**. IEEE, 2005. p. 194–202. ISBN 1424400139. ISSN 1-4244-0013-9. Disponível em: <<http://ieeexplore.ieee.org/document/1542635/>>.

JAGANNATH, J.; SAARINEN, H. M.; DROZD, A. L. Framework for automatic signal classification techniques (FACT) for software defined radios. In: **2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)**. IEEE, 2015. p. 1–7. ISBN 978-1-4673-7557-3. Disponível em: <<http://ieeexplore.ieee.org/document/7208628/>>.

JARARWEH, Y. et al. SD-CRN: Software defined cognitive radio network framework. In: **Proceedings - 2014 IEEE International Conference on Cloud Engineering, IC2E 2014**. IEEE, 2014. p. 592–597. ISBN 9781479937660. Disponível em: <<http://ieeexplore.ieee.org/document/6903534/>>.

JAYAWEERA, S. K. et al. Radiobots: The autonomous, self-learning future cognitive radios. In: **2011 International Symposium on Intelligent Signal Processing and Communications Systems: "The Decade of Intelligent and Green Signal Processing and Communications", ISPACS 2011**. IEEE, 2011. p. 1–5. ISBN 9781457721663. Disponível em: <<http://ieeexplore.ieee.org/document/6146214/>>.

JUANG, B.-H.; RABINER, L. A probabilistic Distance Measure for Hidden Markov Models. **AT&T technical journal**, v. 64, n. 2, p. 391–408, 1985. Disponível em: <https://www.semanticscholar.org/paper/A-probabilistic-distance-measure-for-hidden-Markov-Juang-Rabiner/8b7b9d4439849256c1841f39ff6a38ad93d66a94http://ieeexplore.ieee.org/xpls/abs/{_}all.jsp?arnumber=6770>.

KAMIŃSKI, B.; JAKUBCZYK, M.; SZUFEL, P. A framework for sensitivity analysis of decision trees. **Central European Journal of Operations Research**, Springer Berlin Heidelberg, v. 26, n. 1, p. 135–159, mar 2018. ISSN 16139178. Disponível em: <<http://link.springer.com/10.1007/s10100-017-0479-6>>.

KATIDIOTIS, A.; TSAGKARIS, K.; DEMESTICHAS, P. Performance evaluation of artificial neural network-based learning schemes for cognitive radio systems. **Computers & Electrical Engineering**, v. 36, n. 3, p. 518–535, 2010. ISSN 0045-7906. Disponível em: <http://ac.els-cdn.com/S0045790609001128/1-s2.0-S0045790609001128-main.pdf?{_}tid=6b525f3c-e74b-11e5-b1b9-00000aab0f6c{\&}acdnat=1457674861{_}d494efdc3be12ec64bb6e4b>.

KIM, K. et al. Cyclostationary approaches to signal detection and classification in cognitive radio. In: **2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks**. IEEE, 2007. p. 212–215. ISBN 1424406633. Disponível em: <<http://ieeexplore.ieee.org/document/4221497/>>.

LEVINSON, S. E.; RABINER, L. R.; SONDHI, M. M. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. **Bell System Technical Journal**, v. 62, n. 4, p. 1035–1074, apr 1983. ISSN 15387305. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6768244>>.

LÓPEZ-BENÍTEZ, M.; CASADEVALL, F. A framework for multidimensional modelling of spectrum occupancy in the simulation of cognitive radio systems. In: **2014 9th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2014**. IEEE, 2014. p. 453–458. ISBN 9781479925810. Disponível em: <<http://ieeexplore.ieee.org/document/6923872/>>.

LU, Y. et al. Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks. In: **IEEE Wireless Communications and Networking Conference, WCNC**. IEEE, 2016. v. 2016-Septe, p. 1–6. ISBN 9781467398145. ISSN 15253511. Disponível em: <<http://ieeexplore.ieee.org/document/7564840/>>.

MARINHO, J.; MONTEIRO, E. **Cognitive radio: Survey on communication protocols, spectrum decision issues, and future research directions**. Springer US, 2012. 147–164 p. Disponível em: <<http://link.springer.com/10.1007/s11276-011-0392-1>>.

MARQUES, A. et al. ISCRa - An Intelligent Sensing Protocol for Cognitive Radio. In: **Proceedings - IEEE Symposium on Computers and Communications**. IEEE, 2016. v. 2016-Augus, p. 385–390. ISBN 9781509006793. ISSN 15301346. Disponível em: <<http://ieeexplore.ieee.org/document/7543770/>>.

MARQUES, A. F. F. **SIRCo: uma arquitetura para Sensoriamento Inteligente de Rádios Cognitivos**. Dissertação (Mestrado) — Universidade Federal de Lavras, apr 2016.

MARTINS PINTO, L. R.; CORREIA, L. H. A. **Analysis of machine learning algorithms for spectrum decision in cognitive radios**. [S.l.], 2018. 1–6 p. Disponível em: <<https://ieeexplore.ieee.org/document/8491060/>>.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Kluwer Academic Publishers, v. 5, n. 4, p. 115–133, dec 1943. ISSN 00074985. Disponível em: <<http://link.springer.com/10.1007/BF02478259>>.

MCHENRY, M. A. et al. Chicago spectrum occupancy measurements & analysis and a long-term studies proposal. **Proceedings of the first international workshop on Technology and policy for accessing spectrum - TAPAS '06**, ACM Press, New York, New York, USA, p. 1–es, 2006. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1234388.1234389>>.

MENON, R.; BUEHRER, R. M.; REED, J. H. Outage probability based comparison of underlay and overlay spectrum sharing techniques. In: **2005 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005**. IEEE, 2005. p. 101–109. ISBN 1424400139. Disponível em: <<http://ieeexplore.ieee.org/document/1542623/>>.

MILLER, S. A.; KWONG, W. C. Trellis demodulation of two-level Reed-Solomon-modulated FH-CDMA wireless communication systems. In: **2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT)**. IEEE, 2012. p. 1–6. ISBN 978-1-4577-1343-9. Disponível em: <<http://ieeexplore.ieee.org/document/6223102/>>.

Min Song et al. Dynamic spectrum access: from cognitive radio to network radio. **IEEE Wireless Communications**, v. 19, n. 1, p. 23–29, feb 2012. ISSN 1536-1284. Disponível em: <<http://ieeexplore.ieee.org/document/6155873/>>.

MITOLA, J. Software radios-survey, critical evaluation and future directions. In: [**Proceedings**] **NTC-92: National Telesystems Conference**. IEEE, 1992. p. 13/15–13/23. ISBN 0-7803-0554-X. ISSN 0885-8985. Disponível em: <<http://ieeexplore.ieee.org/document/267870/>>.

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.

ÖNER, M.; JONDRAL, F. On the extraction of the channel allocation information in spectrum pooling systems. **IEEE Journal on Selected Areas in Communications**, v. 25, n. 3, p. 558–565, apr 2007. ISSN 07338716. Disponível em: <<http://ieeexplore.ieee.org/document/4155371/>>.

OSHIRO, T. M.; PEREZ, P. S.; BARANAUSKAS, J. A. How many trees in a random forest? In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.: s.n.], 2012. v. 7376 LNAI, p. 154–168. ISBN 9783642315367. ISSN 03029743.

PANDIT, S.; SINGH, G. Spectrum Sharing in Cognitive Radio using Game Theory. **2013 3rd IEEE International Advance Computing Conference (IACC)**, IEEE, p. 1503–1506, feb 2013. Disponível em: <<http://ieeexplore.ieee.org/document/6514449/http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6514449>>.

PATIL, V. M.; PATIL, S. R. A survey on spectrum sensing algorithms for cognitive radio. In: **2016 International Conference on Advances in Human Machine Interaction, HMI 2016**. IEEE, 2016. p. 149–153. ISBN 9781467388108. ISSN 1553877X. Disponível em: <<http://ieeexplore.ieee.org/document/7449196/>>.

PENG, C.; ZHENG, H.; ZHAO, B. Y. Utilization and fairness in spectrum assignment for opportunistic spectrum access. **Mobile Networks and Applications**, Springer-Verlag New York, Inc., v. 11, n. 4, p. 555–576, aug 2006. ISSN 1383469X. Disponível em: <<http://link.springer.com/10.1007/s11036-006-7322-y>>.

PERKINS, C.; BELDING-ROYER, E.; DAS, S. **Ad hoc On-Demand Distance Vector (AODV) Routing**. IETF, 2003. RFC 3561 (Experimental). (Request for Comments, 3561). Disponível em: <<http://www.ietf.org/rfc/rfc3561.txt>>.

PRINCIPE, J. C.; CHALASANI, R. Cognitive architectures for sensory processing. **Proceedings of the IEEE**, v. 102, n. 4, p. 514–525, apr 2014. ISSN 00189219. Disponível em: <<http://ieeexplore.ieee.org/document/6775277/>>.

QUINLAN, J. R. Learning Efficient Classification Procedures and Their Application to Chess End Games. In: **Machine Learning**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983. p. 463–482. ISBN 978-3-662-12407-9.

QUINLAN, J. R. Induction of Decision Trees. **Machine Learning**, Kluwer Academic Publishers, v. 1, n. 1, p. 81–106, mar 1986. ISSN 15730565. Disponível em: <<http://link.springer.com/10.1007/BF00116251http://dx.doi.org/10.1023/A:1022643204877>>.

RABINER, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. **Proceedings of the IEEE**, v. 77, n. 2, p. 257–286, 1989. ISSN 15582256. Disponível em: <<http://ieeexplore.ieee.org/document/18626/>>.

RENK, T. et al. Bio-inspired algorithms for dynamic resource allocation in cognitive wireless networks. **Mobile Networks and Applications**, Springer US, v. 13, n. 5, p. 431–441, oct 2008. ISSN 1383469X. Disponível em: <<http://link.springer.com/10.1007/s11036-008-0087-8>>.

RIZVI, S.; SHOWANI, N. A Framework to Facilitate Modeling of a Cognitive Radio Network Using Layered Architecture. In: **17th IEEE International Multi Topic Conference 2014**. IEEE, 2014. p. 119–124. ISBN 9781479957545. Disponível em: <<http://ieeexplore.ieee.org/document/7097323/>>.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958. ISSN 0033295X. Disponível em: <<http://doi.apa.org/getdoi.cfm?doi=10.1037/h0042519>>.

RUMELHART, D. E.; MCCLELLAND, J. L.; University of California, S. D. P. R. G. **Parallel distributed processing : explorations in the microstructure of cognition**. [S.l.]: MIT Press, 1986. ISBN 9780262181204.

SAAD, A.; STAEHLE, B.; KNORR, R. Spectrum prediction using hidden Markov models for industrial cognitive radio. In: **International Conference on Wireless and Mobile Computing, Networking and Communications**. [S.l.: s.n.], 2016. ISBN 9781509007240. ISSN 21619654.

SAIFAN, R.; KAMAL, A. E.; GUAN, Y. Spectrum decision for efficient routing in cognitive radio network. In: **MASS 2012 - 9th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems**. IEEE, 2012. p. 371–379. ISBN 9781467324335. Disponível em: <<http://ieeexplore.ieee.org/document/6502537/>>.

SANDRI, M.; ZUCCOLOTTO, P. A bias correction algorithm for the gini variable importance measure in classification trees. **Journal of Computational and Graphical Statistics**, Taylor & Francis, v. 17, n. 3, p. 611–628, sep 2008. ISSN 10618600. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1198/106186008X344522>>.

Shared Spectrum Company. General Survey of Radio Frequency Bands – 30 MHz to 3 GHz. **Shared Spectrum Company Reports**, p. 47, 2010. Disponível em: <www.sharedspectrum.com/wp-content/uploads/2010/0923-General-Band-Survey-30MHz-to-3GHz>

SHARMA, V.; BOHARA, V. Exploiting machine learning algorithms for cognitive radio. In: **Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014**. IEEE, 2014. p. 1554–1558. ISBN 9781479930791. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6968571>>.

SIRIKULVIRIYA, N.; SINTHUPINYO, S. Integration of rules from a random forest. **International Conference on Information ...**, v. 6, p. 194–198, 2011. Disponível em: <<http://www.ipcsit.com/vol6/39-E097.pdf>>.

SONG, Y.; XIE, J. ProSpect: A proactive spectrum handoff framework for cognitive radio Ad hoc networks without common control channel. **IEEE Transactions on Mobile Computing**, v. 11, n. 7, p. 1127–1139, jul 2012. ISSN 15361233. Disponível em: <<http://ieeexplore.ieee.org/document/6202813/>>.

SOUZA, A. D. et al. Experimental Wireless Networking Research using Software-Defined Radios. In: **Anais / XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. [S.l.]: Sociedade Brasileira da Computação, 2017. p. 104–152. ISSN 2177-4978.

STONEHEM, B. **Google Home: An Easy Guide the Features**. CreateSpace Independent Publishing Platform, 2016. (1). ISBN 9781536919783. Disponível em: <<https://books.google.com.br/books?id=Pq3KDAAAQBAJ>>.

SUNDMAN, D. **A Simulator for Cognitive Radio**. Dissertação (Mestrado) — KTH Electrical Engineering, Stockholm, Sweden, 2008.

Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825—2830, 2011.

VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. **IEEE Transactions on Information Theory**, v. 13, n. 2, p. 260–269, apr 1967. ISSN 0018-9448. Disponível em: <<http://ieeexplore.ieee.org/document/1054010/>>.

WANG, G. et al. A comparative assessment of ensemble learning for credit scoring. **Expert Systems with Applications**, Pergamon Press, Inc., v. 38, n. 1, p. 223–230, jan 2011. ISSN 09574174. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S095741741000552X>>.

WIDROW, B.; HOFF, M. **Adaptive switching circuits**. 1960. 96 – 104 p. Disponível em: <<http://www.dtic.mil/docs/citations/AD0241531>>.

YAQUB, M. et al. Improving the classification accuracy of the classic RF method by intelligent feature selection and weighted voting of trees with application to medical image segmentation. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.]: Springer Berlin Heidelberg, 2011. v. 7009 LNCS, p. 184–192. ISBN 9783642243189. ISSN 03029743.

YU, R. et al. Secondary users cooperation in cognitive radio networks: Balancing sensing accuracy and efficiency. **IEEE Wireless Communications**, v. 19, n. 2, p. 30–37, apr 2012. ISSN 15361284. Disponível em: <<http://ieeexplore.ieee.org/document/6189410/>>.

YUCEK, T.; ARSLAM, H. A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications. **Proceedings of the IEEE**, v. 97, n. 5, p. 805–823, 2009. ISSN 1553877X. Disponível em: <http://ieeexplore.ieee.org/document/4796930/http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=4796>.

ZENG, Y. et al. A Review on Spectrum Sensing for Cognitive Radio: Challenges and Solutions. **EURASIP Journal on Advances in Signal Processing**, Springer International Publishing, v. 2010, n. 1, p. 1–16, 2010. ISSN 1687-6172. Disponível em: <<http://asp.eurasipjournals.com/content/2010/1/381465>>.

ZHANG, T.; YU, X. Spectrum Sharing in Cognitive Radio Using Game Theory—A Survey. In: **2010 International Conference on Computational Intelligence and Software Engineering**. IEEE, 2010. p. 1–5. ISBN 978-1-4244-3708-5. Disponível em: <<http://ieeexplore.ieee.org/document/5600846/>>.

ZHAO, J.; ZHENG, H.; YANG, G.-H. Distributed coordination in dynamic spectrum allocation networks. **Proc. IEEE DySpan**, IEEE, p. 259–268, 2005. Disponível em: <<http://ieeexplore.ieee.org/document/1542642/>>.

ZHAO, Q.; SWAMI, A. **A survey of dynamic spectrum access: signal processing and networking perspectives**. [S.l.], 2007.

ZHAO, Q. et al. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. **IEEE Journal on Selected Areas in Communications**, v. 25, n. 3, p. 589–599, apr 2007. ISSN 07338716. Disponível em: <<http://ieeexplore.ieee.org/document/4155374/>>.

ZHENG, H.; CAO, L. Device-centric spectrum management. In: **2005 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005**. IEEE, 2005. p. 56–65. ISBN 1424400139. Disponível em: <<http://ieeexplore.ieee.org/document/1542617/>>.

ZHOU, G.; STANKOVIC, J. A.; SON, S. H. Crowded Spectrum in Wireless Sensor Networks. **Proceedings of Third Workshop on Embedded Networked Sensors (EmNets)**, v. 6, p. 30–31, 2006.