

**DECISÃO DE ESPECTRO EM REDES DE
SENSORES SEM FIO EMPREGANDO
APRENDIZADO DE MÁQUINA**

VINICIUS FONSECA E SILVA

**DECISÃO DE ESPECTRO EM REDES DE
SENSORES SEM FIO EMPREGANDO
APRENDIZADO DE MÁQUINA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: DANIEL FERNANDES MACEDO

Belo Horizonte
Fevereiro de 2014

© 2014, Vinicius Fonseca e Silva.
Todos os direitos reservados.

Silva, Vinicius Fonseca e

S586d Decisão de espectro em redes de sensores sem fio
empregando aprendizado de máquina / Vinicius
Fonseca e Silva. — Belo Horizonte, 2014
xxv, 85 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais

Orientador: Daniel Fernandes Macedo

1. Computação — Teses. 2. Redes de computadores
— Teses. 3. Redes de sensores sem fio — Teses.
4. Aprendizado do computador — Teses. I. Orientador.
II. Título.

CDU 519.6*22(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

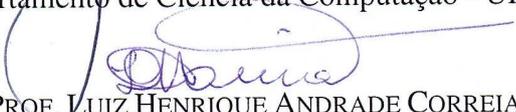
Decisão de espectro em redes de sensores sem fio empregando aprendizado de máquina

VINICIUS FONSECA E SILVA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. DANIEL FERNANDES MACEDO - Orientador
Departamento de Ciência da Computação - UFMG


PROF. JOSÉ MARCOS SILVA NOGUEIRA
Departamento de Ciência da Computação - UFMG


PROF. LUIZ HENRIQUE ANDRADE CORREIA
Departamento de Ciência da Computação - UFLA


PROF. MARCOS AUGUSTO MENEZES VIEIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 21 de fevereiro de 2014.

A meus pais, Almir e Josiara, e minha irmã Fernanda, pelo apoio e confiança incondicionais.

Agradecimentos

A Deus, que me iluminou e me protegeu durante toda essa etapa da minha vida.

A todos da minha família, em especial minha tia Josiane, pelo apoio sempre que precisei.

À Universidade Federal de Minas Gerais e ao Departamento de Ciência da Computação, pela oportunidade de realização do curso de Pós-Graduação.

Aos professores do departamento, pelos conhecimentos adquiridos, os quais levarei pela vida toda.

À CAPES, pelo apoio financeiro concedido durante todo esse período, e que foi de suma importância para o alcance de todos esses objetivos.

A todos os meus amigos e colegas do Laboratório Winet, em especial os amigos Jesse Leoni, Ewerton Salvador, Vinicius Mota, Henrique Moura, Wendley Silva e Romeu Oliveira, que acompanharam de perto o desenvolvimento deste trabalho e me ajudaram diversas vezes, através das várias conversas que tivemos ao longo da pesquisa. Sem essas conversas, inúmeros detalhes decisivos para o andamento do trabalho seriam quase impossíveis de serem solucionados.

Ao meu orientador, professor Daniel Macedo, que sempre me auxiliou nos momentos de dificuldade. Agradeço pela sua paciência ao longo desses dois anos de pesquisa, e acima de tudo, pelas constantes palavras de apoio e de certeza de que este trabalho se concretizaria com sucesso. Agradeço todo o conhecimento e conselhos passados por ele durante os dois anos sob sua orientação, os quais serão de grande valia durante toda a minha vida pessoal e profissional. Sem ele esse trabalho não se concretizaria.

A todos aqueles que não citei aqui, mas que diretamente ou indiretamente contribuíram de alguma forma para a execução deste trabalho.

A todos, os meus sinceros agradecimentos.

*“Não há um homem vivo que não possa
fazer mais do que ele pensa que pode”
(Henry Ford)*

Resumo

Redes de Sensores Sem Fio (RSSFs) utilizam faixas de espectro ISM (Industrial, Scientific and Medical), as quais atualmente encontram-se sobrecarregadas devido a diversas tecnologias tais como WLANs, RSSFs e Bluetooth. Em razão dessa sobrecarga, as RSSFs sofrem com grandes perdas de informação, causando desperdícios de energia elevados. Com o objetivo de reduzir tais perdas, as RSSFs devem empregar métodos inteligentes, como por exemplo de Rádio Cognitivo (RC) para selecionar o melhor canal, coexistindo assim com outras redes que usam a faixa ISM.

Uma das formas de implementação de RC, e que tem sido bastante adotada pela literatura recente, emprega Aprendizado de Máquina (AM), que cria um método de escolha do melhor canal, processo esse que também é conhecido em RC como Decisão de Espectro. Apesar do bom desempenho das propostas existentes, a maior parte possui custo computacional elevado, o que impede seu uso em plataformas com recursos limitados. Outra limitação é o fato da maioria dos trabalhos considerarem um pequeno conjunto de algoritmos de AM. Além disso, grande parte dos trabalhos são validados apenas por meio de simulação e dados empíricos, o que reduz a confiabilidade sobre a eficácia em cenários reais.

Este trabalho avalia um conjunto extenso de algoritmos de AM para a seleção de canal em plataformas de RSSFs de baixo recurso computacional. Os melhores algoritmos de AM são implementados em nós sensores reais, demonstrando experimentalmente sua eficácia. Por fim, apresentamos uma comparação dos algoritmos propostos com métodos baseados em canal fixo, e métodos de Decisão de Espectro baseados em energia. Os resultados obtidos mostram que métodos de RC baseados em AM aumentam o desempenho da comunicação como um todo, quando comparados com os outros métodos implementados, reduzindo a quantidade de retransmissões no meio e assim diminuindo o atraso de envio de pacotes.

Palavras-chave: Rádios Cognitivos, Aprendizado de Máquina, Redes de Sensores Sem Fio.

Abstract

Wireless Sensor Networks (WSNs) use ISM (Industrial, Scientific and Medical) spectrum bands, which are currently overloaded due to various technologies such as WLANs, WSNs and Bluetooth. Because of the overhead, WSNs suffer with great loss of information, causing high waste of energy. Therefore, in order to reduce such losses, WSNs must employ clever methods, such as Cognitive Radio (CR) to select the best channel, coexisting with other networks that use the ISM band.

One of the CR's implementation paradigms, which has been highly adopted by the literature, employs Machine Learning (ML), which creates a method to choose the best channel, process which is also known in the CR area as Spectrum Decision. Despite the good performance of the existing proposals, most of them have high computational costs, which prevent their use on platforms with limited resources. Another limitation is the fact that most of the works consider a small set of ML algorithms. Moreover, the majority of studies are validated only through simulations and empirical data, which reduces the reliability on the effectiveness in real scenarios.

This study evaluates an extensive set of ML algorithms for channel selection in low-resource WSN platforms. The best algorithms are implemented on real sensor nodes, experimentally demonstrating their effectiveness. Finally, we show a comparison of the proposed algorithms with fixed channel and energy-based Spectrum Decision methods. The obtained results show that ML-based RC methods enhance the overall communication performance, when compared with the implemented baselines, reducing the number of retransmissions, and thus reducing the delay.

Keywords: Cognitive Radio, Machine Learning, Wireless Sensor Networks.

Lista de Figuras

1.1	Problema da interferência entre tecnologias na comunicação em RSSFs nas faixas ISM.	2
2.1	Problema do Terminal Escondido em Usuários Primários [Marinho & Monteiro, 2012].	14
2.2	Ciclo ODAL [Jayaweera et al., 2011].	19
4.1	Modelo de predição: versão simplificada.	29
4.2	Modelo de predição: versão detalhada.	32
4.3	Algoritmo de Decisão de Espectro.	33
4.4	Nó sensor Iris [Crossbow, 2008a].	35
4.5	MIB520 [Open Automation, 2013].	35
4.6	RSSF projetada para coleta dos dados de desempenho.	36
5.1	Coefficiente de correlação: medições de todos os nós/tamanho da janela de previsão variável.	54
5.2	Raiz do erro médio quadrático: medições de todos os nós/tamanho da janela de previsão variável.	54
5.3	Atualização da média móvel/tempo decorrido.	55
5.4	Ajuste do valor de α : quantidade de pacotes enviados.	58
6.1	Frequências de operação das redes WLAN e das redes <i>ZigBee</i> [National Instruments, 2013].	72
6.2	Redes WLAN em operação durante a realização dos testes.	73

Lista de Tabelas

4.1	Especificações técnicas para o rádio RF230.	34
5.1	Coefficiente de correlação: medições de um nó/peso variável.	43
5.2	Raiz do erro médio quadrático: medições de um nó/peso variável.	44
5.3	Coefficiente de correlação: medições de um nó/tamanho de base variável.	45
5.4	Raiz do erro médio quadrático: medições de um nó/tamanho de base variável.	46
5.5	Coefficiente de correlação: medições de um nó/tamanho da janela de previsão variável.	48
5.6	Raiz do erro médio quadrático: medições de um nó/tamanho da janela de previsão variável.	49
5.7	Coefficiente de correlação: medições de todos os nós/peso variável.	51
5.8	Raiz do erro médio quadrático: medições de todos os nós/peso variável.	52
5.9	Ajuste do tempo de sensoreamento por canal (em milissegundos).	60
5.10	Teste de sensibilidade: variação no coeficiente de correlação.	63
5.11	Teste de sensibilidade: variação na raiz do erro médio quadrático.	64
6.1	Resultados de desempenho: primeiro teste.	70
6.2	Resultados de desempenho: segundo teste.	71
6.3	Resultados de desempenho: terceiro teste.	73
6.4	Resultados de desempenho: quarto teste.	75
6.5	Consumo de memória para cada solução implementada.	76

Lista de Abreviações

AM	Aprendizado de Máquina
ACK	Acknowledgement
CCA	Clear Channel Assessment
FCC	Federal Communications Commission
ISM	Industrial, Scientific and Medical
KNN	K-Nearest Neighbours
LQI	Link Quality Indication
MLP	Multilayer Perceptron
PAF	Probabilidade de Alarme Falso
PD	Probabilidade de Detecção
POMDP	Partially Observable Markov Decision Process
PU	Primary User
QoS	Quality of Service
RAM	Random Access Memory
RC	Rádio Cognitivo
REP	Reduced Error Pruning
RL	Reinforcement Learning
RNA	Rede Neural Artificial
ROM	Read Only Memory

RSSF	Rede de Sensores Sem Fio
RSSI	Received Signal Strength Indicator
SNIR	Signal to Noise Plus Interference Ratio
SOM	Self Organizing Map
SU	Secondary User
UHF	Ultra High Frequency
VHF	Very High Frequency
WLAN	Wireless Local Area Network

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Abreviações	xxi
1 Introdução	1
1.1 Objetivos	3
1.2 Motivação	4
1.3 Contribuições	4
1.4 Organização do Trabalho	5
2 Referencial Teórico	7
2.1 Rádios Cognitivos	7
2.2 Sensoreamento de Espectro	9
2.2.1 Métodos de Sensoreamento de Espectro	10
2.3 Decisão de Espectro	15
2.4 Aprendizado de Máquina	16
2.5 Conclusão	19
3 Trabalhos Relacionados	21
3.1 Sensoreamento de Espectro	21
3.2 Decisão de Espectro e Aprendizado de Máquina	23
3.2.1 Aprendizado Supervisionado	23

3.2.2	Aprendizado Não Supervisionado	24
3.2.3	Outras Formas de Decisão	25
3.3	Conclusão	26
4	Modelagem Proposta	27
4.1	Seleção de Canal Empregando Aprendizado de Máquina	28
4.2	Modelo de Predição: Visão Geral	29
4.3	Modelo de Predição: Detalhamento	30
4.3.1	Atributos Coletados	30
4.3.2	Modelagem dos Atributos de Entrada e de Saída	31
4.4	Algoritmo de Decisão de Espectro Proposto	32
4.5	Configuração dos Experimentos de Coleta de Dados	34
4.5.1	Plataforma Utilizada	34
4.5.2	Topologia da RSSF para Coleta dos Dados	34
4.5.3	Cenários de Coleta Analisados	35
4.5.4	Parâmetros de Transmissão	36
4.6	Conclusão	37
5	Análise dos Algoritmos de Aprendizado de Máquina	39
5.1	Metodologia de Avaliação dos Algoritmos de AM	39
5.1.1	Algoritmos de AM Testados	40
5.2	Testes Realizados	41
5.2.1	Medições de Um Nó/Peso Variável	41
5.2.2	Medições de Um Nó/Tamanho de Base Variável	42
5.2.3	Medições de Um Nó/Tamanho da Janela de Previsão Variável	45
5.2.4	Medições de Todos os Nós/Peso Variável	47
5.2.5	Medições de Todos os Nós/Tamanho da Janela de Previsão Variável	53
5.3	Ajuste dos Parâmetros de Sensoreamento	55
5.4	Teste de Sensibilidade dos Atributos de Entrada	59
5.4.1	Metodologia dos Testes	61
5.4.2	Resultados	62
5.5	Conclusão	65
6	Avaliação dos Modelos de Predição em uma RSSF Real	67
6.1	Descrição das Implementações	67
6.2	Primeiro Teste	69
6.3	Segundo Teste	70
6.4	Terceiro Teste	71

6.5	Quarto Teste	74
6.6	Consumo de Memória das Soluções Implementadas	76
6.7	Conclusão	76
7	Conclusões	79
7.1	Trabalhos Futuros	80
	Referências Bibliográficas	81

Capítulo 1

Introdução

Um estudo realizado pela FCC (*Federal Communications Commission*), nos Estados Unidos [FCC, 2002], mostrou que as regras atuais adotadas para alocação de canal causam perdas consideráveis na eficiência e aproveitamento do espectro, devido ao fato de que inúmeros intervalos de tempo e espaço do mesmo são raramente utilizados [Yucek & Arslan, 2009; Zeng et al., 2010]. Uma vez que diferentes faixas de frequência são reservadas para um grupo fechado de usuários, os quais possuem uma licença de uso, e assim são ditos usuários licenciados, fica impossibilitado o uso das mesmas por usuários não licenciados, quando estas faixas não estão sendo utilizadas. Este fator alia-se ao crescente número de dispositivos sem fio de diferentes naturezas, além da demanda por uma taxa de transmissão de dados cada vez mais elevada. Com isso, o acúmulo de usuários não licenciados em faixas de frequência livres gera muita interferência, causando grandes perdas de informação e gastos elevados de energia. Uma das redes que mais sofrem com esse problema são as Redes de Sensores Sem Fio (RSSFs), as quais são restritas em energia, já que operam por baterias que não podem ser substituídas na maior parte das vezes, reduzindo assim consideravelmente o seu tempo de vida [Culler et al., 2004].

Por isso, torna-se necessária a adoção de técnicas inovadoras que permitam explorar o espectro com mais eficiência, de forma a possibilitar a coexistência de diferentes redes sem causar conflitos de qualquer natureza entre as mesmas. Os Rádios Cognitivos (RCs) são hoje uma tecnologia promissora, destinada a explorar as faixas de frequência pouco utilizadas, e são considerados como um dos principais alvos de pesquisa em comunicações sem fio [Zeng et al., 2010].

Em faixas licenciadas do espectro, ou seja, em faixas onde deve haver concessões legais para seu uso, há ainda uma classificação que separa os usuários das diferentes redes sem fio em operação em duas categorias distintas [Yucek & Arslan, 2009]. A

primeira delas define os *Usuários Primários*, ou seja, usuários que possuem uma alta prioridade de uso desses canais. Três exemplos clássicos de Usuários Primários são as estações de TV, estações de rádio e as redes celulares. A segunda categoria define os *Usuários Secundários*, os quais possuem menor prioridade de operação sobre as faixas licenciadas, e que devem explorar o espectro sem causar conflitos sobre Usuários Primários ou outros Usuários Secundários. Um exemplo de Usuário Secundário que é bastante utilizado na atualidade são os microfones sem fio, os quais normalmente operam nas faixas VHF (*Very High Frequency*) ou UHF (*Ultra High Frequency*), utilizadas por algumas estações de TV ao redor do mundo.

As RSSFs, pelo fato de operarem somente nas faixas ISM (*Industrial, Scientific and Medical*), não possuem classificação de usuário. Porém, essa classe de rede está sujeita na maior parte do tempo a interferências causadas por usuários de redes Wi-Fi e redes Bluetooth, as quais também operam nas faixas ISM. O problema da coexistência das RSSFs com outras redes, o qual é exemplificado graficamente na Figura 1.1, é um dos principais focos de estudo deste trabalho. Nesta Figura, dois nós pertencentes a uma RSSF têm sua comunicação prejudicada por várias colisões, em razão de outras redes estarem operando na mesma faixa de frequência e no mesmo intervalo de tempo.

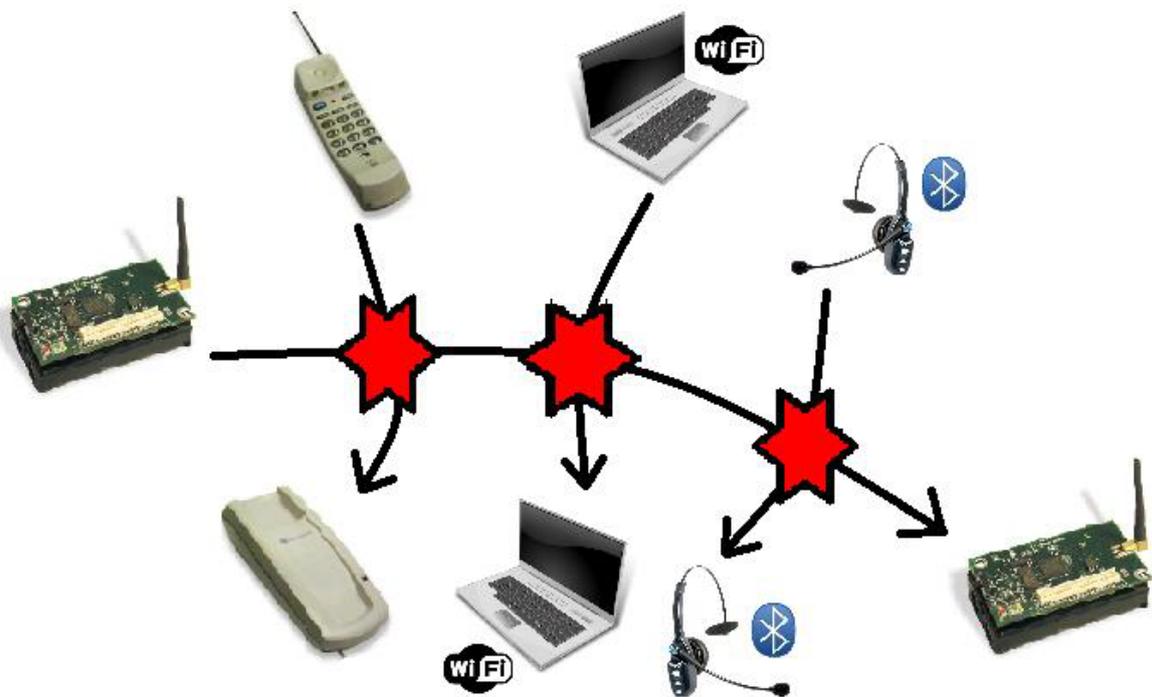


Figura 1.1. Problema da interferência entre tecnologias na comunicação em RSSFs nas faixas ISM.

Segundo Akyildiz et al. [2009], as características de um RC podem ser divididas

em quatro partes. O *Sensoreamento de Espectro* obtém informações de uma série de canais, detectando os intervalos de tempo não utilizados e em melhores condições de uso pelos RCs. A *Decisão de Espectro* analisa as informações obtidas no Sensoreamento e escolhe os melhores canais com base em métricas de Qualidade de Serviço (*Quality of Service - QoS*). Já o *Compartilhamento de Espectro* aloca os canais selecionados e coordena as transmissões entre os diversos usuários da rede. Por fim, a *Mobilidade de Espectro*, em geral estima as condições futuras da rede, identificando a necessidade de troca de canal.

Em geral, os RCs utilizam-se de uma gama extensa de informações relativas ao canal em que estão operando. Como consequência, parte dos métodos propostos na literatura recente apresentam alta complexidade, o que pode causar demanda por uma grande quantidade de recursos computacionais. Este fator, por sua vez, pode inviabilizar o uso dos métodos propostos em RSSFs, pelo fato de possuírem limitações de memória e processamento, além de limitações de energia como mencionado anteriormente.

A alta complexidade de parte dos métodos propostos oferece também o risco da indisponibilidade de algumas das informações utilizadas. A disponibilidade de informações muitas vezes depende da plataforma utilizada, mais especificamente do modelo de rádio utilizado, o que pode também inviabilizar ou não o uso de tais métodos. Portanto, um dos desafios atuais na implementação de RCs em RSSFs é a proposta de métodos que disponham de um conjunto de informações mais limitado e abstrato do canal, mas que independa da plataforma utilizada, ou seja, que esteja disponível em qualquer plataforma, ao mesmo tempo em que possa ser coletado de forma simplificada.

1.1 Objetivos

O objetivo principal deste trabalho é propor algoritmos para RC em RSSFs para as fases de Sensoreamento e Decisão de Espectro, de baixo custo computacional. Tais algoritmos tomarão como base um preditor obtido a partir de algoritmos de Aprendizado de Máquina (AM), cuja entrada será um conjunto de informações simplificado e abstrato do canal. O uso de AM concentrará todo o processamento e definição do conhecimento do meio sensoreado na fase de Decisão, fazendo com que a etapa de Sensoreamento seja apenas uma etapa de coleta de informações sobre os canais.

Os algoritmos desenvolvidos serão aplicados no contexto de RSSFs, pelo fato dessa classe de rede ter apresentado um grande crescimento de interesse por parte da comunidade acadêmica nos últimos anos. Acredita-se que o presente trabalho poderá

servir de base para futuras pesquisas na área de RCs, por englobar diversos algoritmos em um único contexto.

Para isso, o desempenho de uma gama extensa de algoritmos de AM será avaliada por meio de uma mesma ferramenta de análise, utilizando dados provenientes de uma RSSF real. Além disso, os modelos de predição obtidos nos melhores algoritmos de AM serão aplicados em plataformas reais, avaliando o seu desempenho prático em diferentes cenários.

1.2 Motivação

Devido aos recursos limitados de hardware e energia da maioria das plataformas de RSSFs existentes no mercado, fica evidente a dificuldade de implementação de RCs de baixa complexidade e que não prejudiquem o fluxo de dados úteis. Além disso, tem-se observado uma escassez de trabalhos com implementação em plataformas reais, como apontado por Yarkan et al. [2011b]. Isto ocorre principalmente pelo fato de que poucas delas oferecem suporte necessário para se implementar o algoritmo em sua totalidade, o que pode inviabilizar a execução de testes mais completos e confiáveis.

O uso de AM tem sido uma prática cada vez mais comum em trabalhos que propõem algoritmos de Sensoreamento e Decisão de Espectro. Entretanto, grande parte desses trabalhos focam em um pequeno conjunto de algoritmos de AM. Além disso, em razão da complexidade de tempo e espaço elevadas de alguns dos algoritmos de AM, grande parte das soluções propostas na literatura são avaliadas apenas por meio de simulação. Cada autor analisa os algoritmos propostos seguindo metodologias diversas, utilizando ferramentas de análise diferentes, o que oferece o risco de discrepâncias quanto à confiabilidade dos resultados obtidos. Por fim, outro fator é que boa parte utiliza bases de dados artificiais na fase de testes, obtidas por *software*, e que podem não representar o mundo real com detalhamento suficiente.

1.3 Contribuições

Em suma, as principais contribuições presentes no trabalho proposto são apresentadas abaixo.

- Análise de um extenso número de algoritmos de AM, de diversas classes, disponíveis em uma mesma ferramenta de análise, considerando-se as limitações de *hardware* inerentes às RSSFs;

- Implementação dos melhores algoritmos em nós sensores reais, avaliando o impacto no desempenho teórico obtido pela ferramenta de análise, sob diferentes cenários;
- Comparação das soluções implementadas com soluções baseadas em canal fixo e de seleção de canal com base em energia, as quais são bastante utilizadas na literatura.

1.4 Organização do Trabalho

O presente trabalho está organizado da seguinte forma. No Capítulo 2 são apresentados os conceitos empregados na dissertação. No Capítulo 3 são mostrados os principais trabalhos relacionados, apontando as diferenças e limitações em relação às soluções aqui propostas. Em seguida, no Capítulo 4 é apresentada a modelagem do problema, com uma descrição da solução a ser implementada em nós sensores reais, e uma descrição da coleta dos dados para utilização nos algoritmos de AM. No Capítulo 5, por sua vez, é apresentada a fase de avaliação dos algoritmos de AM, definindo-se os melhores parâmetros para a construção dos preditores segundo a modelagem adotada. Os melhores preditores obtidos pelos algoritmos de AM são implementados em uma RSSF real e seu desempenho é analisado no Capítulo 6 sob diferentes cenários. O Capítulo 7 conclui o trabalho e descreve os trabalhos futuros.

Capítulo 2

Referencial Teórico

Neste capítulo são apresentados os conceitos de RC envolvidos neste trabalho. Inicialmente é feita uma conceituação geral de RC e suas principais características, seguida de um detalhamento daquelas que serão o foco do tema aqui proposto. Por fim, é feita uma apresentação dos conceitos envolvendo AM e sua aplicação em RC.

As seções deste capítulo estão divididas da seguinte forma. A Seção 2.1 faz uma conceituação geral de RCs e suas principais características. As seções 2.2 e 2.3 apresentam uma descrição detalhada do funcionamento das etapas de Sensoreamento e Decisão de Espectro, respectivamente, com uma divisão bem definida dos tipos de sensoreamento existentes. A Seção 2.4 mostra os principais conceitos envolvendo AM e seus objetivos principais, seguidos de sua contextualização em RC. A Seção 2.5 fecha o capítulo, com algumas considerações sobre os temas aqui apresentados.

2.1 Rádios Cognitivos

Existem diversas variações e significados para o termo “*Rádio Cognitivo*”, dependendo do contexto e aplicação do mesmo. A definição mais adequada para o contexto deste trabalho e que é adotado pela FCC é apresentada a seguir: “*Rádio Cognitivo é um rádio ou sistema que mensura o ambiente eletromagnético no qual ele opera, podendo ajustar seus parâmetros operacionais de forma autônoma e dinâmica, de forma a maximizar o fluxo de dados, mitigar a interferência, facilitar a interoperabilidade e acessar mercados secundários*” [FCC, 2005].

Como visto no Capítulo 1, os usuários presentes em uma faixa licenciada do espectro se dividem em duas categorias principais: *Usuários Primários* e *Usuários Secundários*. Os *Usuários Primários* possuem licença para operarem na faixa de frequência em questão, ou seja, possuem permissão prévia para comunicar entre si a qualquer mo-

mento e durante qualquer intervalo de tempo. Já os *Usuários Secundários* não possuem essa licença, e por isso, é necessário que o meio seja sensorado de forma a verificar se existem *Usuários Primários* ou outras redes compostas por *Usuários Secundários* em operação, de forma a evitar interferências entre eles.

O propósito principal de um RC é o reuso ou o compartilhamento do espectro. Caso o objetivo principal no uso do mesmo seja o aumento na eficiência de um canal licenciado, *Usuários Secundários* devem constantemente verificar a ocorrência de atividades por parte de *Usuários Primários*. Uma vez detectada essa ocorrência, a qual deve ser feita em tempo hábil, o canal deve ser liberado durante o tempo necessário. Caso o objetivo seja a coexistência de múltiplos usuários em canais não reservados, o qual se encaixa no contexto do presente trabalho, deve-se possibilitar a convivência destes dispositivos, ao mesmo tempo garantindo uma comunicação eficiente e confiável.

Como também visto no Capítulo 1, as características de um RC são divididas em quatro partes distintas: *Sensoreamento de Espectro*, *Decisão de Espectro*, *Compartilhamento de Espectro* e *Mobilidade de Espectro* [Akyildiz et al., 2009]. Tais características são descritas com mais detalhes a seguir.

- *Sensoreamento de Espectro (Spectrum Sensing)*: Consiste da obtenção de informações sobre um conjunto de canais, de forma a detectar os intervalos de tempo não utilizados, bem como os canais menos utilizados e portanto mais propícios ao uso pelos RCs. Essas informações podem ser variáveis físicas provenientes do ambiente, bem como informações de desempenho da rede, provenientes de camadas superiores da pilha de protocolos.
- *Decisão de Espectro (Spectrum Decision)*: Se resume na análise das informações obtidas na fase de *Sensoreamento de Espectro*, com base em um ou mais critérios, que variam de acordo com a aplicação e ambiente de operação, de forma a determinar os melhores canais a serem utilizados. Ainda de acordo com Akyildiz et al. [2009], nessa tarefa preocupa-se também com a construção das rotas, uma vez que a alocação de canais pode variar em pontos distintos.
- *Compartilhamento de Espectro (Spectrum Sharing)*: Realiza alocação dos canais selecionados e coordena as transmissões entre os diversos usuários, de forma a minimizar as colisões entre eles e reduzir ao máximo a interferência causada sobre os *Usuários Primários*.
- *Mobilidade de Espectro (Spectrum Mobility)*: Akyildiz et al. [2009] define como principal objetivo nessa tarefa, em faixas licenciadas do espectro, a requisição de uso do mesmo por um *Usuário Primário*, forçando assim os *Usuários Secundários*

a mudarem para outra faixa livre, processo também conhecido como *Handoff de Espectro*. Em faixas não licenciadas, pode-se definir aqui como objetivo principal a percepção de uma degradação do desempenho ou da confiabilidade dos enlaces, tornando a troca de canal necessária.

As características de RC que serão o foco deste trabalho, e que serão abordadas com mais detalhes nas seções a seguir são o *Sensoreamento de Espectro* e a *Decisão de Espectro*, pelo fato de se encaixarem de forma adequada aos conceitos de AM utilizados neste trabalho. Além disso, neste contexto, o *Sensoreamento de Espectro* torna-se apenas uma fase de coleta de informações do meio, como dito anteriormente, deixando assim a previsão e escolha dos melhores canais para a etapa de Decisão.

2.2 Sensoreamento de Espectro

A ideia principal do Sensoreamento de Espectro é obter um ou mais parâmetros do ambiente no qual a rede está operando, com o objetivo de analisar o estado de um ou mais canais em um determinado intervalo de tempo. Esses parâmetros podem ser tanto físicos, os quais podem ser obtidos por meio de sensores, como de desempenho obtidos por software ao longo do tempo. Em outras palavras, o Sensoreamento de Espectro procura obter um perfil de utilização do mesmo em diversos domínios, tais como no tempo, no espaço e na frequência. O nível de complexidade do sensoreamento varia de método para método, o que pode influenciar na confiabilidade e precisão das informações obtidas. Em alguns métodos, são consideradas características inerentes ao sinal recebido, tais como a modulação, amplitude e a largura de banda, por exemplo. Apesar de oferecer maior confiabilidade sobre a análise realizada, a obtenção desses parâmetros requer técnicas mais elaboradas de análise de sinal, o que incrementa a complexidade computacional [Yucek & Arslan, 2009]. Portanto, dependendo da plataforma de *hardware* sendo utilizada, o uso dessas técnicas pode ser inviável. Ansari & Mähönen [2010] enfatizam a inviabilidade de se utilizar os métodos de sensoreamento propostos a redes sem fio tradicionais em RSSFs, devido à sua alta complexidade e à demanda por uma grande quantidade de informações do meio, o que não é possível nas RSSFs devido às limitações de tempo e energia.

O Sensoreamento de Espectro é considerado hoje uma das tarefas mais desafiadoras na área de RCs, devido a uma série de fatores. Um dos principais está associado à Relação Sinal Ruído Mais Interferência (*Signal to Noise Plus Interference Ratio - SNIR*), necessária para se diferenciar sinais provenientes de outros usuários do ruído proveniente do ambiente. Caso essa relação seja reduzida, ou seja, caso o sinal re-

cebido seja fraco em relação ao ruído proveniente do ambiente, a detecção pode não ser possível. Boa parte dos trabalhos existentes na literatura demonstram a eficácia dos algoritmos propostos por meio de modelagens numéricas e simulações, porém, os mesmos cenários podem não acontecer em aplicações reais, o que pode invalidar o método proposto. Além disso, modelar o meio sem fio em uma simulação é uma tarefa geralmente complicada, a qual corre o risco de não fazer uma representação fiel da realidade.

Outro fator que torna o Sensoreamento de Espectro desafiador é a variação na acurácia do método adotado, em razão da topologia e ambiente locais. Liu et al. [2010] mostraram que um modelo de sensoreamento criado em uma das entidades da rede sofre uma perda considerável na acurácia quando utilizado em outra entidade com localização e perfil de tráfego variados.

Por fim, uma característica que pode impactar consideravelmente na acurácia do sensoreamento é a definição correta dos parâmetros a serem sensoreados, bem como a quantidade de amostras necessária e como as mesmas devem ser coletadas. Tsagkaris et al. [2012] testaram o método proposto em ambiente real para conjuntos de entrada e números de amostras diferentes, sendo considerados parâmetros provenientes de diferentes camadas da pilha de protocolos. Tamma et al. [2012] propõem ainda um grafo que auxilia na escolha do melhor método de amostragem, de acordo com a aplicação desejada, o que indiretamente pode auxiliar no conjunto de parâmetros mais adequado. Como o foco deste trabalho são as RSSFs, o ideal é que o conjunto de entrada seja o menor possível, tanto em número de amostras como em número de parâmetros, permitindo reduzir o tempo e a complexidade do algoritmo de sensoreamento, e aumentando o tempo de vida da rede.

Em vista dos desafios apresentados acima, o Sensoreamento de Espectro se tornou uma área de pesquisa bastante abordada nos últimos anos [Yucek & Arslan, 2009; Zeng et al., 2010]. Vários métodos de sensoreamento de espectro foram propostos na literatura, sendo que os principais serão abordados com mais detalhes nas seções a seguir.

2.2.1 Métodos de Sensoreamento de Espectro

Os diferentes algoritmos de Sensoreamento de Espectro propostos na literatura podem ser divididos em duas categorias principais: Sensoreamento Local e Sensoreamento Cooperativo. No sensoreamento local, cada usuário realiza uma análise do espectro independentemente, criando assim seu próprio modelo. Já no Sensoreamento Cooperativo, a ideia principal é aumentar a precisão e confiabilidade do sensoreamento realizado

através do compartilhamento de informações mensuradas por cada um dos usuários da rede, distribuídos em grande quantidade e de forma ampla no ambiente [Marinho & Monteiro, 2012; Song et al., 2012; Yu et al., 2012]. Cada uma das duas categorias será apresentada com mais detalhes a seguir.

2.2.1.1 Sensoreamento de Espectro Local

Grande parte dos algoritmos de Sensoreamento de Espectro local propostos na literatura tendem a partir de um dos três “pilares”: *Energy Detection*, *Matched Filter Detection* e *Feature Detection* [Song et al., 2012; Marinho & Monteiro, 2012]. O método *Energy Detection* terá um foco maior neste trabalho, pelo fato de ser simplificado e facilmente aplicável no mundo real [Khaleel et al., 2012]. Além disso, este método está altamente relacionado com um dos propósitos deste trabalho, que é a criação de algoritmos que sejam de fácil aplicação em RSSFs, e sem que os recursos computacionais disponíveis sejam fortemente comprometidos.

Energy Detection

Nesse método, é realizada uma amostragem da intensidade de sinal (*RSSI - Received Signal Strength Indicator*), cujo valor é comparado com um limiar (*threshold*) pré definido, sendo considerado como um dos métodos mais simples de ser implementado [Marinho & Monteiro, 2012; Khaleel et al., 2012]. Esse método se baseia em duas hipóteses, aqui denominadas de H_0 e H_1 :

$$\begin{aligned} H_0 &: r(t) \\ H_1 &: s(t) + r(t) \end{aligned}$$

A hipótese H_0 indica a presença apenas de ruído, aqui denominado $r(t)$, representando assim um canal desocupado. Já a hipótese H_1 representa a presença de Usuário Primário no sinal recebido, aqui representado por $s(t)$, mais o ruído proveniente do ambiente. Esse ruído pode ser modelado de diversas formas, dentre elas, a mais utilizada é o Ruído Branco Gaussiano (*White Gaussian Noise*), por ter uma representação mais simplificada.

Com base nessas hipóteses, grande parte dos trabalhos avalia o método de sensoreamento baseado em energia em termos de duas probabilidades: a Probabilidade de Alarme Falso (PAF) e a Probabilidade de Detecção (PD). A PAF indica a chance de se concluir H_1 , ou seja, a detecção de um usuário na rede, sendo que na realidade o canal está desocupado (ou seja, H_0), desperdiçando assim oportunidades no uso do espectro. Já a PD indica a chance de se concluir H_1 , sendo que o canal está de fato

ocupado. Portanto, um algoritmo de sensoreamento ideal procura minimizar PAF ao mesmo tempo em que maximiza PD.

Um dos principais problemas enfrentados nesse método é o risco de incertezas com a relação sinal-ruído, o que pode muitas vezes impedir a detecção de Usuários Primários [Song et al., 2012]. Isso dificulta a definição correta, ao longo do tempo, do limiar que separa o ruído do ambiente de um Usuário Primário, como forma de reduzir ao máximo a PAF [Marinho & Monteiro, 2012; Yucek & Arslan, 2009]. A fixação do limiar em um valor pré-definido é aqui considerada uma tarefa arriscada, principalmente quando o algoritmo desenvolvido é aplicado em cenários onde o nível de tráfego das redes, assim como o ruído do ambiente, varia constantemente ao longo do tempo.

Outros dois problemas relacionados a esse método são apontados no trabalho de Akyildiz et al. [2011]. O primeiro deles é a definição do tempo de sensoreamento. Existem algumas técnicas, tais como a apresentada por Tamma et al. [2012], que auxiliam na definição do tempo de sensoreamento que procure maximizar a acurácia na predição. Caso a relação sinal-ruído seja pequena, esse tempo tende a ser mais elevado, de forma que a detecção seja feita de forma mais precisa. O segundo problema está relacionado à distinção de sinais provenientes de Usuários Primários de sinais provenientes de Usuários Secundários, o que não é possível em métodos baseados em energia, pelo fato de que aqui é possível distinguir apenas o sinal de um dos dois tipos de usuários do ruído do ambiente. Com isso, também é necessário que haja uma sincronização entre os Usuários Secundários, para que esses não transmitam informações entre si durante o sensoreamento. O Sensoreamento Cooperativo busca solucionar esse problema, o qual será abordado com mais detalhes na seção 2.2.1.2.

Matched Filter Detection

A principal característica presente nesse método é a existência de um conhecimento prévio do perfil do sinal produzido pelos Usuários Primários, ou seja, diversas características como o tipo de modulação, frequência e formato dos pacotes de dados produzidos por esses usuários são reconhecidas pelos Usuários Secundários desde o início de sua operação, o que reduz consideravelmente o tempo de sensoreamento [Song et al., 2012; Yucek & Arslan, 2009].

A principal limitação presente é a necessidade de rádios que consigam demodular o sinal recebido de Usuários Primários, de tal forma que seja possível extrair todas as características apresentadas acima. A classificação do sinal recebido de forma quase exata exige a aplicação de diferentes algoritmos de recepção, o que aumenta considera-

velmente o consumo de energia e a complexidade dos algoritmos desenvolvidos [Yucek & Arslan, 2009]. Algumas plataformas como Mica2 [Crossbow, 2003], MicaZ [Crossbow, 2008b], Iris [Crossbow, 2008a] e TmoteSky [MoteIv, 2006]), largamente utilizadas em RSSFs não oferecem, em seu conjunto de *hardware* padrão, suporte direto à obtenção de todas essas informações, e por isso uma possível implementação pode vir a ser cara.

Feature Detection

Este método parte do princípio de que os sinais seguem um padrão periódico, ou seja, parâmetros estatísticos como média e autocorrelação variam periodicamente ao longo do tempo [Song et al., 2012; Yucek & Arslan, 2009]. Em outras palavras, a cicloestacionariedade (*cyclostationarity*), a qual é obtida por meio de uma função de correlação espectral, é bem definida em sinais gerados pelos Usuários Primários, ao passo que o ruído do ambiente é sempre estacionário, não possuindo correlação espectral [Lee et al., 2008; Zeng et al., 2010]. Por isso, uma das principais vantagens obtidas com o seu uso é a realização de uma distinção bem definida entre sinal e ruído, uma das principais dificuldades encontradas no método *Energy Detection*. Além disso, ele tem a facilidade de distinguir entre diferentes tipos de sinais, sendo robusto a alarmes falsos provenientes de sinais de outros Usuários Secundários ou de interferências [Lee et al., 2008].

Apesar da alta precisão obtida na detecção de sinais de Usuários Primários através desses parâmetros, existem algumas desvantagens no uso desse método. Uma das principais é a alta taxa de amostragem necessária para se determinar a cicloestacionariedade, o que aumenta consideravelmente a complexidade computacional, tornando-se inviável em aplicações com grandes restrições de tempo [Zeng et al., 2010].

2.2.1.2 Sensoreamento de Espectro Cooperativo

Em regiões isoladas da rede, inúmeros efeitos adversos provenientes do meio, tais como o enfraquecimento do sinal gerado pelos Usuários Primários, fazem com que o Sensoreamento de Espectro Local seja ineficiente em alguns momentos [Song et al., 2012]. Por isso, o Sensoreamento de Espectro Cooperativo é proposto como uma alternativa para aumentar a precisão na detecção, através do compartilhamento de informações sobre o espectro entre múltiplos Usuários Secundários.

Uma das vantagens nesse tipo de sensoreamento é a eliminação de um problema bastante conhecido em redes sem fio, denominado como *Problema do Terminal Escondido*. Segundo Marinho & Monteiro [2012], esse problema impediria em alguns casos a

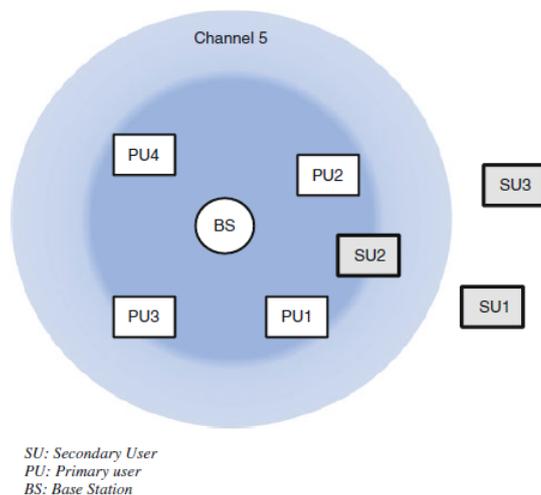


Figura 2.1. Problema do Terminal Escondido em Usuários Primários [Marinho & Monteiro, 2012].

detecção de Usuários Primários. Na Figura 2.1, onde é esquematizado esse problema, os Usuários Secundários $SU1$ e $SU3$ detectam o canal 5 como desocupado, pois estão fora do alcance dos Usuários Primários $PU1$, $PU2$, $PU3$ e $PU4$, que utilizam esse mesmo canal. Porém, o Usuário Secundário $SU2$ não se encontra nessa condição, e reporta a ocupação do canal 5 pelos PUs a $SU1$ e $SU3$.

É possível ainda, através da cooperação entre Usuários Secundários, definir a região de alcance dos Usuários Primários, através de alguns resultados de sensoriamento incorretos apresentados ao longo do tempo [Yang et al., 2010]. Essa definição poderá, portanto, analisar possíveis ocorrências do Problema do Terminal Escondido ao longo do tempo.

Atualmente, a literatura divide o Sensoriamento de Espectro Cooperativo em dois paradigmas: cooperação entre Usuários Primários e Usuários Secundários, e cooperação apenas entre Usuários Secundários [Yu et al., 2012]. No primeiro, uma vez que os Usuários Primários têm conhecimento da presença de Usuários Secundários, é definida uma política de uso das faixas do espectro licenciadas, de forma a evitar conflitos e melhorar a qualidade de comunicação. Já no segundo paradigma não existe interação entre esses dois tipos de usuários.

Em linhas gerais, como é mostrado por Yu et al. [2012], o Sensoriamento de Espectro Cooperativo se divide em duas etapas principais:

- *Seleção dos usuários que farão o sensoriamento:* Uma vez que um dos Usuários Secundários deseja enviar informações a outro Usuário Secundário, o primeiro envia uma requisição ao *Centro de Fusão* de dados, o qual será responsável por

coordenar todo o processo de sensoriamento. O *Centro de Fusão* por sua vez seleciona um subconjunto de Usuários Secundários, os quais serão responsáveis por analisar o espectro. Denkovski et al. [2011] propõem o uso de Mapas Ambientais de Rádio (*Radio Environmental Maps*), os quais fazem uma inferência da distribuição dos nós da rede, auxiliando na escolha do melhor conjunto de nós para sensoriamento.

- *Encaminhamento dos resultados ao Centro de Fusão*: Segundo Zeng et al. [2010], as informações a serem retornadas ao *Centro de Fusão* podem ser de dois tipos principais. O primeiro deles consiste de decisões individuais feitas por cada um dos usuários selecionados, com base no sensoriamento realizado por cada um deles. Em outras palavras, cada usuário selecionado definirá localmente os melhores canais, os quais serão encaminhados para o *Centro de Fusão*. Feito isso, o *Centro de Fusão* deverá combinar essas informações com base em algum critério, como a localização por exemplo, para se tomar a decisão final, a qual será encaminhada ao restante da rede. O segundo tipo de informação consiste apenas dos dados de sensoriamento, sem decisão prévia, a qual será feita unicamente pelo *Centro de Fusão*. Esse segundo tipo de compartilhamento é utilizado no trabalho de Khaleel et al. [2010], onde as informações de sensoriamento são combinadas e os canais são avaliados com base em uma função de probabilidade.

Um dos grandes desafios atuais no uso de Sensoriamento de Espectro Cooperativo é a definição da quantidade de informações que poderão ser trocadas entre as entidades da rede, em qual momento isso deverá acontecer, bem como quais nós da rede deverão realizar o sensoriamento. A principal justificativa para isso é que o compartilhamento dessas informações causa um aumento no tráfego da rede, o que pode elevar consideravelmente a latência e a probabilidade de colisões, aumentando a necessidade de retransmissões e causando assim um gasto desnecessário de energia. Além disso, a acurácia do sensoriamento pode variar de acordo com a localização, como mencionado anteriormente [Akyildiz et al., 2009].

2.3 Decisão de Espectro

Uma vez obtidas as faixas de frequência disponíveis através da fase de Sensoriamento de Espectro, é necessário decidir ou não pela troca de canal de uma ou mais entidades da rede, bem como a redefinição de outros parâmetros de configuração inerentes à pilha de protocolos. Essa etapa é chamada de *Decisão de Espectro*.

Assim como na fase de Sensoreamento, na fase de Decisão podemos ter tanto uma abordagem distribuída, onde cada nó constrói sua própria base de conhecimento e toma decisões de forma independente, como uma abordagem centralizada, onde dados de sensoreamento são encaminhados a um coordenador, o qual tomará uma decisão que será encaminhada de forma unificada ou independente a cada uma das entidades da rede. Felice et al. [2011] procuram resolver o problema de alocação de espectro, onde o aprendizado é compartilhado entre as entidades da rede, de forma que as mesmas confrontem diversos resultados e criem uma base de conhecimento mais confiável, em uma tentativa de eliminação da interferência causada nos resultados obtidos em intervalos anteriores.

A definição de um modelo estatístico simplificado de uso do espectro por Usuários Primários, proposto por Zhao & Swami [2007], pode servir também de auxílio em aplicações onde não é possível obter informações do meio em tempo real. Além disso, a coleta de informações sobre o espectro ao longo do tempo pode auxiliar na predição do uso do mesmo em intervalos futuros, podendo maximizar ainda mais o ganho obtido nas decisões [Marinho & Monteiro, 2012]. Essa coleta de informações sinaliza a possibilidade de adaptação dos diferentes algoritmos de AM existentes na literatura, de forma que possam ser utilizados para predição de uso do canal em RC, tópico esse que será abordado nas seções a seguir.

Uma das principais questões apontadas nessa etapa é a heterogeneidade do espectro ao longo de múltiplos saltos desde um usuário de origem até um destinatário, o que dificulta a seleção de uma rota ótima sem levar em conta a seleção de canais realizada. A realização do roteamento e da seleção de canais de forma independente, como proposto por Wang & Zheng [2006], se mostra uma técnica pouco eficiente no contexto de RC, pois ainda está propensa a interferir na comunicação de Usuários Primários em um ou mais saltos na rota selecionada. Portanto, os protocolos de roteamento existentes devem ser revistos e adaptados de forma a considerarem as informações resultantes do sensoreamento. Além disso, a redefinição do protocolo de roteamento e de outros protocolos da pilha podem influenciar diretamente no atraso fim-a-fim e na taxa de transmissão dos dados da rede, como mostrado por Akyildiz et al. [2009].

2.4 Aprendizado de Máquina

O Aprendizado de Máquina consiste em um conjunto de técnicas de manipulação de dados com o objetivo de detectar padrões nos mesmos, definindo regras de decisão que representem esses padrões. Em outras palavras, o uso de AM permite compreender de

forma simplificada os efeitos de uma massa de dados sobre uma métrica pré-definida, auxiliando na modelagem correta desses dados e com isso otimizando as regras de decisão produzidas [Clancy et al., 2007]. Dessa forma, reduz-se o erro das decisões em relação ao mundo real, cuja magnitude é altamente dependente da aplicação desejada.

Em suma, existem três abordagens em AM [Katidiotis et al., 2010]: *Supervisionado*, *Não-Supervisionado* e *Semi-Supervisionado*. O *Aprendizado Supervisionado* busca construir as regras de decisão com base em um conjunto de exemplos conhecidos e bem definidos, diretamente ligados ao mundo real, sendo esse conjunto chamado de *Base de Treinamento*. Cada um desses exemplos, denominados *Instâncias*, são compostos por um conjunto de valores de entrada e um respectivo valor de saída. Tais valores podem ser do tipo nominal, o qual é representado por uma palavra ou símbolo pré-definido pelo usuário de acordo com o problema abordado, também chamado de *Classe*, ou do tipo numérico, o qual é representado por um intervalo no conjunto dos números reais ou inteiros, e que pode ser finito ou não.

Já o *Aprendizado Não Supervisionado* não dispõe de uma base inicial de conhecimento, de forma que o sistema otimiza, durante a sua operação, as suas regras de decisão. Esse tipo de aprendizado toma apenas como base dados brutos ou pré-processados, provenientes do mundo real, organizados em um conjunto de valores de entrada mas que não estão associados a um determinado valor de saída, como visto no *Aprendizado Supervisionado*. Com isso, nesta forma de aprendizado, é feita uma busca por similaridades entre esses dados brutos, os quais permitam definir uma classe ou associar esses dados a um mesmo intervalo numérico, permitindo assim a criação das regras de decisão.

Por fim, o *Aprendizado Semi-Supervisionado* realiza o aprendizado tanto com uma base de treinamento, antes da ativação do sistema, quanto com cada execução do sistema em operação. Em outras palavras, ele é uma espécie de junção entre os dois tipos de aprendizado vistos acima, onde o sistema é iniciado com um modelo pré-definido, podendo ser modificado com a inclusão de novos dados em tempo de execução.

Em AM, existem dois tipos principais de algoritmos: *Classificação* e *Regressão*. Os algoritmos de *Classificação* têm seu valor de saída definido por um conjunto de classes finito, de fácil interpretação e criadas pelo usuário de acordo com o problema abordado. Não existem limites teóricos para o número de classes criadas, entretanto, estão altamente ligadas ao problema abordado, o qual em um contexto prático pode limitar o número de classes em razão da carência de recursos computacionais, como visto nas RSSFs.

Já os algoritmos de *Regressão* têm seu valor de saída definido por um valor nu-

mérico, situado em um intervalo no conjunto dos números reais e que pode ser finito ou não. Neste caso, um dos fatores principais que podem limitar o uso desses algoritmos é a precisão de valores desejada, ou seja, a diferença entre o valor aproximado e o valor real, análoga ao erro de classificação em um classificador. Quanto maior a precisão, maior será o consumo de memória e processamento. As unidades de ponto flutuante, responsáveis pelo gerenciamento dessa precisão, são representadas por um módulo em *hardware* destinado exclusivamente a essa função. Entretanto, essas unidades podem não estar presentes em alguns casos, o que causa a necessidade de uma emulação dessa função em *software*, aumentando o tempo de processamento dos dados e consequentemente do algoritmo de regressão. O mesmo vale para os algoritmos de *Classificação*, quando estes trabalham com o tipo numérico nos valores de entrada.

Para ambos os tipos de algoritmos citados acima, podemos ter conjuntos de valores de entrada homogêneos, ou seja, conjuntos com apenas um dos tipos citados anteriormente, ou heterogêneos, que misturam os dois tipos. Para ambos os casos, é importante observar que estes também possuem as limitações de aplicação prática mencionadas anteriormente, as quais devem ser vistas com cuidado ao se definir os valores de entrada para o problema abordado.

Uma das principais vantagens no uso de algoritmos de AM em RCs é a sua capacidade de modelar o comportamento da rede em intervalos futuros, permitindo a aplicação de soluções em tempo hábil, como a troca de canal [Bantouna et al., 2012]. Em aplicações onde não é possível obter informações do meio em tempo real, a definição de um modelo estatístico simplificado de uso do espectro por outros usuários do mesmo meio, como proposto por Zhao & Swami [2007], é uma forma alternativa para o processo de decisão.

Por outro lado, uma das principais desvantagens no uso de AM é o risco da demanda por uma grande quantidade de recursos computacionais, em razão da estrutura de dados utilizada ou da forma como os dados são manipulados de forma a obter a previsão. Alguns algoritmos de AM, por exemplo, necessitam de que toda a base de treinamento seja armazenada de forma que a mesma possa ser consultada ou modificada em tempo real, o que pode ser inviável em uma RSSF com capacidade de armazenamento reduzida. Além disso, tratamentos de dados complexos, em alguns casos, podem fazer com a obtenção da previsão seja um processo demorado, impactando negativamente na aplicação em execução. Há também a impossibilidade de determinar se os modelos criados pelos algoritmos de AM são ótimos, e se de fato atendem a todos os casos de uso analisados.

No contexto de RC, o objetivo principal é a otimização de métricas de Qualidade de Serviço (*Quality of Service - QoS*), ou parâmetros provenientes de uma ou mais



Figura 2.2. Ciclo ODAL [Jayaweera et al., 2011].

camadas da pilha de protocolos. Uma métrica bastante utilizada é o *bitrate*, o qual determina o montante de dados trafegados ao longo de um intervalo de tempo [Bantouna et al., 2012]. Um *bitrate* maior indicará um fluxo de dados maior, trazendo uma melhor experiência ao usuário final. Entretanto nem todos os rádios possuem *bitrate* variável, como é no caso das RSSFs, o que impede o uso do mesmo como fator de otimização.

Jayaweera et al. [2011] propuseram o conceito “Observar-Decidir-Agir-Aprender” (*Observe, Decide, Act, Learn - ODAL*), esquematizado na Figura 2.2, definindo de forma precisa os RCs do futuro. Nesse trabalho, eles afirmam que os rádios terão a capacidade de aprender com os erros do passado e assim convergir a uma configuração ótima, além de considerar uma série de parâmetros como atrasos de comunicação, consumo de energia, acurácia de sensoriamento, segurança, dentre outros. A experiência adquirida pelo RC na etapa de Decisão, a qual está diretamente ligada ao conceito ODAL, possibilita que as ações a serem tomadas em intervalos futuros possam ser otimizadas, fato este que pode ser alcançado com o uso de AM.

2.5 Conclusão

Este capítulo apresentou os principais conceitos de RCs envolvidos no presente trabalho. Foram apresentados os diversos tipos de *Sensoreamento de Espectro* presentes nos RCs tradicionais, com foco principal no método de Sensoreamento baseado em energia (*Energy Detection*), além de uma breve descrição da fase de *Decisão de Espectro*. Por fim, foi feita uma introdução de AM e sua aplicação em RCs.

Uma das vantagens obtidas com o uso de métodos estatísticos, como o *Feature Detection* apresentado na Seção 2.2.1.1, é a baixa dependência de informações do ambiente, como o RSSI (*Received Signal Strength*), por exemplo, o que pode contribuir

para a redução do tempo de sensoriamento. Porém, de acordo com Zeng et al. [2010] sua principal desvantagem ainda é sua vulnerabilidade a interferências inesperadas, que fogem consideravelmente da modelagem realizada.

Já no Sensoriamento Cooperativo, algumas questões como a definição de um canal de controle, a seleção dos usuários para realização do sensoriamento em redes cuja topologia muda através do tempo, a sincronização desses usuários e a segurança das informações trafegadas são consideradas como os principais desafios ao se desenvolver um método desse tipo [Yu et al., 2012]. A construção e atualização de rotas em aplicações onde múltiplos canais estarão sendo utilizados, bem como a notificação de mudança de canal por cada um dos usuários também são algumas das questões a serem consideradas.

Capítulo 3

Trabalhos Relacionados

Este capítulo apresenta alguns dos trabalhos relacionados, relativos às fases de RCs consideradas no Capítulo 2. Na Seção 3.1, são mostrados trabalhos que procuram atacar alguns dos principais problemas relativos à fase de Sensoreamento de Espectro em RCs, focando exclusivamente nos métodos baseados em energia (*Energy Detection*). Em seguida, a Seção 3.2 traz trabalhos recentes relativos à aplicação de AM na etapa de Decisão de Espectro, sob as diferentes abordagens de AM apresentadas na Seção 2.4, bem como outras formas de Decisão existentes na literatura recente. A Seção 3.3 fecha o capítulo com as conclusões e algumas considerações.

3.1 Sensoreamento de Espectro

Como visto no Capítulo 2, um dos principais problemas existentes nos métodos de Sensoreamento baseados em energia (*Energy Detection*) é a separação de sinais de Usuários Primários do ruído do meio. Uma das formas de separação é por meio do ajuste de um limiar, que separa ambos em tempo de execução. Kim et al. [2010] propuseram o uso de histogramas, ao passo que Song et al. [2011] utiliza tabelas pré-definidas para se determinar o valor mais apropriado para o limiar. Shi & de Francisco [2011] optam pelo embasamento em métricas como a ocupação do canal, Balieiro et al. [2010], por sua vez, tomam com base parâmetros estabelecidos pelo padrão IEEE 802.22, padrão esse criado exclusivamente para RCs.

Outros trabalhos definem o limiar com base em premissas que garantam que a PAF seja inferior a um valor pré-determinado [Stabellini & Javed, 2010]. Song et al. [2011] utilizam componentes de *hardware* destinados exclusivamente ao isolamento de sinais de Usuários Primários, permitindo realizar o ajuste do limiar de forma mais confiável. Entretanto, o uso de tais componentes pode aumentar consideravelmente o

custo financeiro da plataforma utilizada, consequentemente inviabilizando o uso dessa técnica de sensoriamento em redes de larga escala. Por fim, Taj & Akil [2011] aproveitaram da cicloestacionariedade para prever a evolução do espectro ao longo do tempo, utilizando Redes Neurais Artificiais.

Yarkan et al. [2011a] realizaram estudos sobre os efeitos da linha de visada, ou seja, do contato visual entre dois RCs, na presença e ausência de informações sobre o espectro em ambientes *indoor*. Foram também analisados os efeitos da ausência de linha de visada e do fenômeno de *shadowing*, que é a flutuação da intensidade de sinal recebido em razão dos obstáculos presentes no meio. Dentre os resultados obtidos, observou-se que quando o RC possui linha de visada, não há necessidade de uso de um método de Sensoriamento de Espectro complexo, uma vez que é garantido que o desempenho de métodos simples como o *Energy Detection* será satisfatório.

Uma das formas de se inferir se os RCs se encontram com ou sem linha de visada é por meio dos Mapas Ambientais de Rádio (*Radio Environmental Maps - REMs*), apresentado na Seção 2.2.1.2. Através deles, é possível selecionar os nós com melhor localização para realizar o Sensoriamento, aumentando a confiabilidade sobre a saída obtida e reduzindo os custos de processamento e energia. Meshkova et al. [2011] descrevem um *testbed* para criação de REMs em ambientes *indoor*. Os resultados obtidos, por meio de experimentos práticos em cenários distintos, demonstraram a necessidade de que os REMs sejam criados e modificados dinamicamente em tempo real, devido à heterogeneidade do espectro.

No trabalho de Pal & Nasipuri [2012], foi proposto um algoritmo de seleção de canal para RSSFs, o qual foi testado em nós sensores reais, com o objetivo de reduzir o *overhearing*. O *overhearing* é um fenômeno relativo à recepção de informações destinadas a outros nós da rede, causando gastos desnecessários de energia. Os autores utilizam uma métrica baseada no consumo de energia dos nós, determinando os canais mais adequados para transmissão. Apesar de não estar diretamente ligado ao Sensoriamento de Espectro, é uma métrica que pode ser usada para reduzir a complexidade do algoritmo em instantes da rede que se sabe que usuários primários não estão presentes.

No contexto de coexistência entre redes em frequências não licenciadas, Ansari et al. [2011] desenvolveram o *WiSpot*, uma plataforma de RSSFs composta por dois rádios 802.15.4 (padrão *ZigBee*) com o objetivo de detectar usuários de redes 802.11. O sensoriamento é feito de forma sincronizada e aproveita, dentre outras informações, de interrupções do rádio como a SFD (*Start Frame Delimiter - Delimitador de Início de Quadro*) para diferenciar transmissões 802.11 de transmissões 802.15.4, o que ajuda na simplificação do método proposto. Como visto em trabalhos acima, a principal limitação presente é necessidade de componentes de *hardware* adicionais, no caso o

segundo rádio, o que aumenta o custo financeiro da plataforma.

Ainda em um contexto de coexistência, Stabellini & Javed [2010] buscam analisar experimentalmente duas abordagens: no domínio do tempo, onde apenas um canal é sensoreado, e no domínio da frequência, onde uma série de canais é sensoreada, realizando assim o *ranking* e seleção do melhor canal. Ambas as abordagens utilizam somente o RSSI (*Received Signal Strength Indicator*) como atributo para determinar o quão ocupado está o canal sensoreado. Os autores concluem que soluções de RC com abordagem no domínio da frequência, apesar do *overhead* adicional com o sensoreamento realizado, promovem ganhos de desempenho por meio de um melhor aproveitamento do espectro.

3.2 Decisão de Espectro e Aprendizado de Máquina

A identificação dos canais livres na etapa de Sensoreamento é considerada aqui como parte da inteligência agregada pelo rádio ao longo do tempo. Entretanto, quando utilizam-se técnicas de AM para escolha do canal, a tendência é que toda a inteligência do rádio passe a se concentrar na etapa de Decisão, tornando o Sensoreamento apenas uma etapa de coleta de informações do meio e/ou do canal, como dito anteriormente.

3.2.1 Aprendizado Supervisionado

Em *Aprendizado Supervisionado*, uma das técnicas mais utilizadas devido à sua modelagem relativamente simples são as Redes Neurais Artificiais (RNAs), onde o conhecimento é armazenado na forma de pesos, que são atribuídos às conexões entre os neurônios. No trabalho desenvolvido por Tsagkaris et al. [2008], são propostas duas estruturas de RNA para predição do *bitrate*: uma básica e outra estendida, com a diferença de que a estendida considera uma gama de informações maior que a básica. A comparação entre as soluções mostrou que a acurácia da predição pode mudar consideravelmente em função dos parâmetros considerados, bem como em função do meio sensoreado. Um exemplo disso pode ser visto na RNA desenvolvida por Baldo et al. [2009], onde são adicionados parâmetros temporais, considerando assim a variabilidade do tráfego ao longo do dia e da semana. Outro exemplo pode ser visto no trabalho desenvolvido por Liu et al. [2010] onde foram construídos três RNAs distintas, cuja diferença se encontrava na escala de tempo utilizada para os valores de entrada.

Com base nestes trabalhos, procuramos reunir a maior quantidade possível de parâmetros para construção da base de treinamento a ser utilizada nos algoritmos de AM. Entretanto, uma preocupação adicional e que não foi observada de forma direta nos

trabalhos citados anteriormente foi a de que esses parâmetros pudessem ser coletados em tempo hábil, além de serem facilmente interpretáveis, o que aqui são considerados fatores essenciais para a obtenção de modelos de predição mais simplificados.

Outro fator que influencia na acurácia da predição em algoritmos utilizando RNA, e que está diretamente relacionada à escolha adequada dos parâmetros de treinamento é a sua topologia, a qual se separa entre duas categorias distintas [Katidiotis et al., 2010]: Redes *Feed-Forward*, onde os dados passam por diversas camadas em sequência até produzir uma ou mais saídas, e Redes Recorrentes, as quais possuem uma ou mais saídas de uma camada conectada às entradas da mesma ou de camadas anteriores, dando o sentido de sensibilidade a eventos anteriores. Uma das topologias mais utilizadas na literatura é a MLP (*Multi-Layer-Perceptron*), devido à facilidade de modelagem e implementação oferecida pela mesma. Em algumas aplicações torna-se necessário o uso de uma grande quantidade de neurônios para que um MLP possa ter uma acurácia aceitável, o que pode aumentar exponencialmente o tempo de predição, devido à grande quantidade de operações matemáticas exigidas. Além disso, uma grande quantidade de conexões implica que uma grande quantidade de pesos terá que ser armazenada, exigindo assim uma alocação de memória extensa, o que pode não ser possível em algumas plataformas comerciais de RSSFs [Liu & Cerpa, 2011].

Em Liu & Cerpa [2011] é feita uma comparação entre três técnicas de previsão, procurando determinar previamente a sua implementação em uma plataforma comercial de RSSFs. Entretanto, seu foco é na estimação da qualidade da comunicação entre nós vizinhos, de forma a modificar o roteamento, enquanto no presente trabalho estima-se a qualidade da comunicação de uma série de canais disponíveis ao rádio com o objetivo de escolher o melhor canal.

3.2.2 Aprendizado Não Supervisionado

Em *Aprendizado Não Supervisionado*, existem poucos trabalhos voltados para a área de RCs. Um dos mais recentes é o desenvolvido por Tsagkaris et al. [2012], onde são apresentados os Mapas Auto-Organizáveis (*Self-Organizing Maps* - SOMs). Esse trabalho é desenvolvido de forma pioneira, uma vez que os SOMs ainda não haviam sido utilizados no contexto de RCs. A ideia principal é coletar amostras multi-dimensionais, e a partir delas construir uma rede composta por vários neurônios, cujas distâncias entre si variam de acordo com a sua similaridade. Neurônios muito próximos entre si definem um *cluster*, o qual representa uma determinada classificação. A definição inicial da topologia é feita com base em um conjunto de dados pré-classificados, de forma a descobrir quantos neurônios serão necessários, bem como a quantidade de *clusters*.

Tsagkaris et al. [2012] utilizam os SOMs para predição do *bitrate*, demonstrando ser uma técnica relativamente simples de modelagem e implementação.

Entretanto, a principal desvantagem observada foi a grande quantidade de memória necessária para sua execução, uma vez que os SOMs podem ser compostos por uma grande quantidade de neurônios, dependendo da aplicação executada. Além disso, é necessário que uma análise extensiva dos neurônios do SOM seja realizada a cada previsão, para que o valor ou classe mais adequada seja encontrada dado um conjunto específico de valores de entrada. O presente trabalho vai no sentido contrário dessa proposta, adotando algoritmos de AM que permitam realizar uma previsão em tempo hábil, sem a necessidade de uma consulta extensiva a valores previamente armazenados.

3.2.3 Outras Formas de Decisão

Dentro do escopo de AM, outras formas de aprendizado pouco utilizadas também são vistas na literatura recente. Unnikrishnan & Veeravalli [2010] definem o sensoreamento e acesso ao espectro como um Processo de Decisão de Markov Parcialmente Observável (*Partially Observable Markov Decision Process - POMDP*). Em outras palavras, o uso dos canais não é conhecido de forma explícita pelos Usuários Secundários, seguindo assim uma evolução Markoviana. Uma vez que o usuário acessa um canal livre, o mesmo recebe uma recompensa proporcional à largura de banda do canal acessado. De forma contrária, o canal sofre penalização caso esteja ocupado.

Alguns trabalhos propõem soluções utilizando Aprendizado por Reforço (*Reinforcement Learning - RL*). Através das ações realizadas no passado, realiza-se todo um processo de otimização, aumentando a chance de sucesso sobre as ações futuras. Felice et al. [2010] apresentam um conjunto de técnicas envolvendo RL que podem ser aplicadas em RCs.

A principal desvantagem observada nas formas de aprendizado apresentadas acima é a dependência de funções de recompensa e penalização que sejam altamente precisas. Muitas RSSFs, pelo fato de possuírem *hardware* limitado, não possuem fatores de precisão elevados o suficiente para trabalhar com algumas dessas funções, que algumas vezes exigem cálculos complexos e de custo computacional elevado. Por isso, uma quebra na precisão pode prejudicar o desempenho dessas funções, inviabilizando seu uso em RSSFs. Este trabalho tem o diferencial de que procura adotar técnicas de AM que, ainda com a precisão reduzida, podem oferecer desempenho adequado.

Fora do escopo de AM, é possível encontrar na literatura recente diferentes propostas de métodos de Decisão de Espectro para seleção de canal. No trabalho de Masonta et al. [2013], são apresentados e descritos os estudos mais recentes de RCs

para esse propósito. São também apresentados estudos recentes relacionados à caracterização do canal, considerada pelos autores como uma etapa anterior à seleção de canal, e reconfiguração, considerada uma etapa posterior, ambas inerentes à Decisão de Espectro. Os autores fazem uma divisão entre trabalhos com propostas a redes de RCs centralizadas e redes de RCs não centralizadas.

Ainda segundo Masonta et al. [2013], uma das principais questões de pesquisa em aberto é a necessidade de cooperação entre SUs, de forma a explorar a diversidade espacial do canal analisado, ao mesmo tempo em que o *overhead* causado por essa cooperação seja reduzido. Outra questão é a frequente troca de canal em redes de RCs com múltiplos saltos. Uma vez que a seleção de canal em diferentes regiões tende a ser distinta, haverá atrasos na transmissão causados por trocas de canal ao longo da rota. Por isso, há a necessidade de se desenvolver protocolos de roteamento em conjunto com os métodos de Decisão de Espectro, de forma a reduzir esses atrasos.

No contexto de RSSFs, Correia et al. [2012] propõem dois métodos de Decisão de de Espectro para seleção de canal em RSSFs. O primeiro deles toma como base medições de RSSI e resultados apresentados pelo algoritmo CCA (*Clear Channel Assessment*) em cada canal analisado. Já o segundo método toma como base diferentes critérios, tais como informações provenientes da camada física e requisitos da aplicação em execução, os quais são ponderados por meio de entropia.

3.3 Conclusão

Este capítulo apresentou os principais trabalhos relacionados ao tema aqui proposto, com foco em Sensoreamento e Decisão de Espectro. Tendo em vista as diferentes propostas apresentadas, algumas considerações devem ser realizadas. A primeira delas está relacionada ao nível de complexidade presente na fase de Sensoreamento de Espectro, o qual tende a ser elevado como pôde ser visto em alguns dos métodos apresentados na Seção 3.1. Portanto, este é considerado atualmente um dos principais desafios ao se colocar em prática um método de Sensoreamento proposto, em qualquer tipo de rede [Zeng et al., 2010].

A dificuldade de implementação dos métodos propostos em uma plataforma real, em razão do impacto da mesma no desempenho apresentado por meio de simulação, tem se mostrado uma preocupação cada vez mais comum na atualidade. Por isso, fica evidente a necessidade de uma análise prévia desses métodos considerando-se as limitações de *hardware* inerentes à plataforma na qual eles serão aplicados, para que o impacto no desempenho seja o mínimo possível.

Capítulo 4

Modelagem Proposta

Neste capítulo é apresentada a modelagem do problema aqui abordado. Esta etapa precede a análise de desempenho dos algoritmos de AM, a qual será descrita com mais detalhes no Capítulo 5.

As seções neste capítulo estão organizadas da seguinte forma. A Seção 4.1 mostra como a seleção do melhor canal, com vistas à melhora no desempenho da rede, pode ser realizada com o uso de AM. As seções de 4.2 a 4.4 apresentam a modelagem do problema do ponto de vista teórico, por meio de uma abordagem *bottom-up*, mostrando passo-a-passo a construção do algoritmo de Decisão de Espectro aqui proposto. A Seção 4.2 apresenta uma descrição simplificada do preditor criado para compor esse algoritmo. A Seção 4.3 faz um detalhamento desse preditor, por meio da descrição dos atributos utilizados e a modelagem adotada para os mesmos. Em seguida, a Seção 4.4 descreve o funcionamento do algoritmo de Decisão de Espectro, o qual será composto pelo preditor e será implementado em nós sensores reais.

Uma vez feita a modelagem teórica do problema, a Seção 4.5 apresenta a modelagem do ponto de vista prático, descrevendo a configuração dos experimentos de coleta da massa de dados bruta. Nesta seção são mostrados detalhes da plataforma de RSSFs utilizada, bem como os parâmetros de transmissão e os cenários de coleta analisados. Os dados aqui coletados serão processados segundo a modelagem teórica feita na seção 4.3, e irão compor as bases de treinamento a serem aplicadas sobre os algoritmos de AM, cuja análise de desempenho será descrita com mais detalhes no Capítulo 5. A Seção 4.6 fecha o capítulo com as conclusões.

4.1 Seleção de Canal Empregando Aprendizado de Máquina

Uma vez que desejamos selecionar o melhor canal dentre um conjunto de candidatos, com o objetivo final de melhorar o desempenho da rede como um todo, é necessário fazer um ranqueamento dos mesmos. Uma das formas de se realizar esse procedimento é por meio de previsões. A principal vantagem em se realizar uma previsão é a possibilidade de se fazer uma análise do fenômeno de forma mais estendida, a médio ou a longo prazo, classificando-o de forma mais confiável.

Uma previsão pode ser realizada sob diferentes pontos de vista, dependendo da aplicação desejada. No contexto deste trabalho, desejamos selecionar o canal que proporcione o melhor desempenho da rede, portanto, definimos como previsão o desempenho da rede em intervalos futuros. Esta previsão irá tomar como base um conjunto de atributos, os quais são altamente correlacionados com essa previsão. Tais atributos podem, por exemplo, ser resultado de uma sequência de observações, sobre um fenômeno do ambiente ou até mesmo características operacionais da rede ao longo do tempo, definindo assim um histórico dessas observações. No nosso caso, como a previsão é feita sobre o desempenho da rede em intervalos futuros, definimos como um conjunto de atributos o desempenho da rede em intervalos anteriores, estabelecendo um alto nível de correlação entre esses valores, como dito anteriormente.

Com base nas escolhas acima, definimos portanto um oráculo, o qual responde à seguinte pergunta: *Para um determinado canal, com base no desempenho da rede obtido em intervalos anteriores, qual será o desempenho da mesma no futuro?*

Uma vez que esta pergunta é respondida para cada um dos canais candidatos, ou seja, são realizadas previsões para cada um dos mesmos, é selecionado o canal que possuir a melhor previsão, ou seja, que apresentar o melhor desempenho da rede no futuro. Com isso, a rede deverá mudar para esse canal de forma que o desempenho seja maximizado.

A realização de uma previsão com base em um conjunto de dados específica é facilmente alcançado com o uso de AM. Inicialmente devemos definir um preditor, o qual irá deter todo o conhecimento a respeito do comportamento do meio. Em seguida, devemos especificar quais atributos serão coletados e qual a melhor modelagem a ser adotada sobre os mesmos, de forma a serem aplicados posteriormente no preditor. Por fim, é necessário definir a forma como esse preditor será utilizado, criando assim o algoritmo de Decisão de Espectro a ser implementado em nós sensores reais. Todos esses passos serão descritos com mais detalhes nas seções a seguir.

4.2 Modelo de Predição: Visão Geral

Como visto na Seção 2.4, os algoritmos de AM, por meio da manipulação de uma massa de dados pré-definida, buscam padrões e assim constroem uma série de regras de decisão que representem esses padrões, os quais definem uma classe ou um valor numérico. Essas regras, agrupadas segundo uma estrutura ditada pelo algoritmo de AM aplicado, definem um modelo que generaliza a massa de dados trabalhada, o qual é denominado *Modelo de Predição*.

Com isso, a partir de uma determinada amostra coletada em tempo real, e composta por um conjunto de informações chamadas de *Atributos de Entrada*, é gerada uma *Saída*, segundo a generalização feita pelo modelo de predição construído.

No contexto de RCs em RSSFs aqui trabalhado, definimos como atributos de entrada um histórico, representado pelo desempenho sob diversas métricas das últimas transmissões e recepções realizadas, e como valor de saída o desempenho dos pacotes a serem transmitidos, segundo o *Número de Tentativas de Transmissão Total*. Com isso, temos um modelo de predição que busca aproximar o desempenho futuro da rede, em termos do total de tentativas de transmissão, tomando apenas como base o comportamento da RSSF em intervalos anteriores.

Portanto, quanto maior o valor de saída, ou seja quanto maior o número de tentativas de transmissão total estimado pelo modelo de predição, maior será o tempo gasto nas transmissões, e maior será o nível de ocupação do canal em questão. O canal selecionado será aquele cuja previsão apresentar o menor número de tentativas de transmissão total. A Figura 4.1 apresenta uma representação gráfica simplificada do preditor.

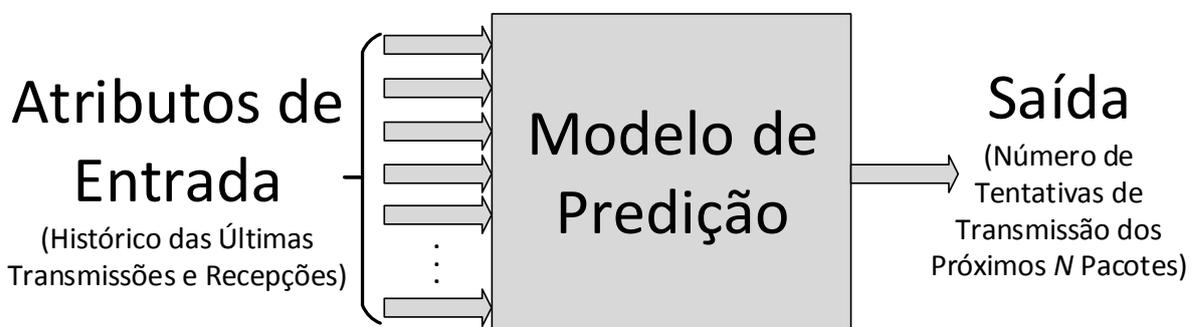


Figura 4.1. Modelo de predição: versão simplificada.

4.3 Modelo de Predição: Detalhamento

Nesta seção é feita uma descrição detalhada dos atributos coletados, os quais representam a massa de dados mencionada anteriormente, bem como a modelagem adotada, a qual definiu os atributos de entrada utilizados no modelo de predição.

4.3.1 Atributos Coletados

Os seguintes atributos foram coletados para construção dos atributos de entrada dos algoritmos de AM:

- **RSSI (*Received Signal Strength Indicator*)**: Medido em dBm , quantifica a força com a qual o sinal do meio é recebido pelo rádio em um determinado instante. Para este atributo, é coletada uma quantidade fixa de amostras imediatamente antes de cada transmissão realizada.
- **Número de Tentativas de Transmissão Total**: Quantidade de tentativas realizadas para se transmitir um pacote, com sucesso ou não, a um determinado destinatário da rede.
- **Contagens das Razões de cada Tentativa Realizada**: Em caso de uma ou mais falhas na transmissão de um determinado pacote, também serão contabilizadas as suas razões, definidas abaixo:
 1. **ERX**: O rádio estava ocupado com a recepção de outro pacote;
 2. **ECCA**: O algoritmo CCA, implementado em *hardware*, indicou que o canal estava ocupado (detecção de portadora);
 3. **ENACK**: Falha ocorrida em razão do rádio ter transmitido um pacote e a sua confirmação (pacote *ACK - Acknowledgement*) não ter sido recebida. Esta métrica captura as colisões, que podem tanto ocorrer no quadro de dados quando na confirmação. Desta forma, ocorrendo uma colisão, o nó sensor não recebe o quadro de ACK.
- **Dados dos últimos pacotes recebidos**: São coletados a cada pacote recebido o seu RSSI e o LQI (*Link Quality Indication*). O LQI é uma métrica definida pelo padrão IEEE 802.15.4, também chamado de padrão *ZigBee*, que mensura a qualidade do enlace no instante de recepção do pacote, em termos do nível de distorção do sinal recebido. Ao contrário do RSSI coletado antes das transmissões, este possui maior granularidade ($1\ dBm$), em razão do rádio utilizado.

4.3.2 Modelagem dos Atributos de Entrada e de Saída

Para a construção das bases de treinamento, adotou-se o conceito de média móvel ponderada. O uso de média móvel ponderada nos atributos de entrada define um histórico do desempenho das últimas transmissões, designando um menor peso às transmissões mais antigas e reduzindo a variação na entrada do preditor. Com isso, reduzimos o número de atributos de entrada, simplificando o modelo de predição a ser construído. A Equação 4.1 define o cálculo da média móvel V no instante t (V_t), onde E_t é o valor de um dos atributos de entrada listados anteriormente no instante t , e V_{t-1} é o valor da média móvel no instante $t - 1$.

$$V_t = V_{t-1} \times (1 - \alpha) + E_t \times \alpha \quad (4.1)$$

O valor de α determina a rapidez com a qual a entrada reage a uma mudança no seu sinal (por exemplo um aumento no valor de entrada). Para evitar que o preditor tome uma decisão com base em informações que não condizem com a realidade, o meio deve ser sensoreado diversas vezes.

Para a saída, optamos por uma abordagem de regressão, ou seja, foi escolhido um valor numérico, representado pela variável *Número de Tentativas de Transmissão dos próximos N pacotes* como dito anteriormente, e que é aqui denominada de *Janela de Previsão*, onde N é um parâmetro que será ajustado posteriormente. Esta variável foi escolhida pois o seu valor indica a qualidade do meio tendo em vista tanto interferências externas (ruídos do ambiente, interferências de outras redes no mesmo espectro) como internas (transmissões de dados no mesmo canal). Um canal que transmite a informação com menor quantidade de tentativas ocupa o meio por menos tempo, e portanto deverá ser um canal com menor atraso e taxa de perda.

Outro motivo para a escolha de um valor numérico para a saída é o fato de que, para o contexto de RSSFs aqui considerado, optamos por um preditor que trabalhasse com valores mais granularizados, ou seja, que permitisse um ajuste fino desses valores. Uma abordagem de classificação, por outro lado, não oferece esse ajuste, e em razão da alta variabilidade do meio, oferece o risco de um desempenho insatisfatório. Isso foi comprovado por meio de testes preliminares, os quais serão omitidos por razões de simplificação dos resultados apresentados.

Nestes testes, foi utilizada a mesma variável (*Número de Tentativas de Transmissão dos próximos N pacotes*) na forma de classes pré-definidas. Os resultados obtidos não foram satisfatórios, uma vez que os classificadores detectaram corretamente as instâncias de uma das classes, ao passo que as outras não apresentaram boa separação. Além disso, foi também observado que os modelos de classificação gerados eram

consideravelmente extensos, inviabilizando sua implementação em uma RSSF real.

A Figura 4.2 apresenta uma representação gráfica do preditor, agora com os atributos de entrada segundo a modelagem de média móvel.

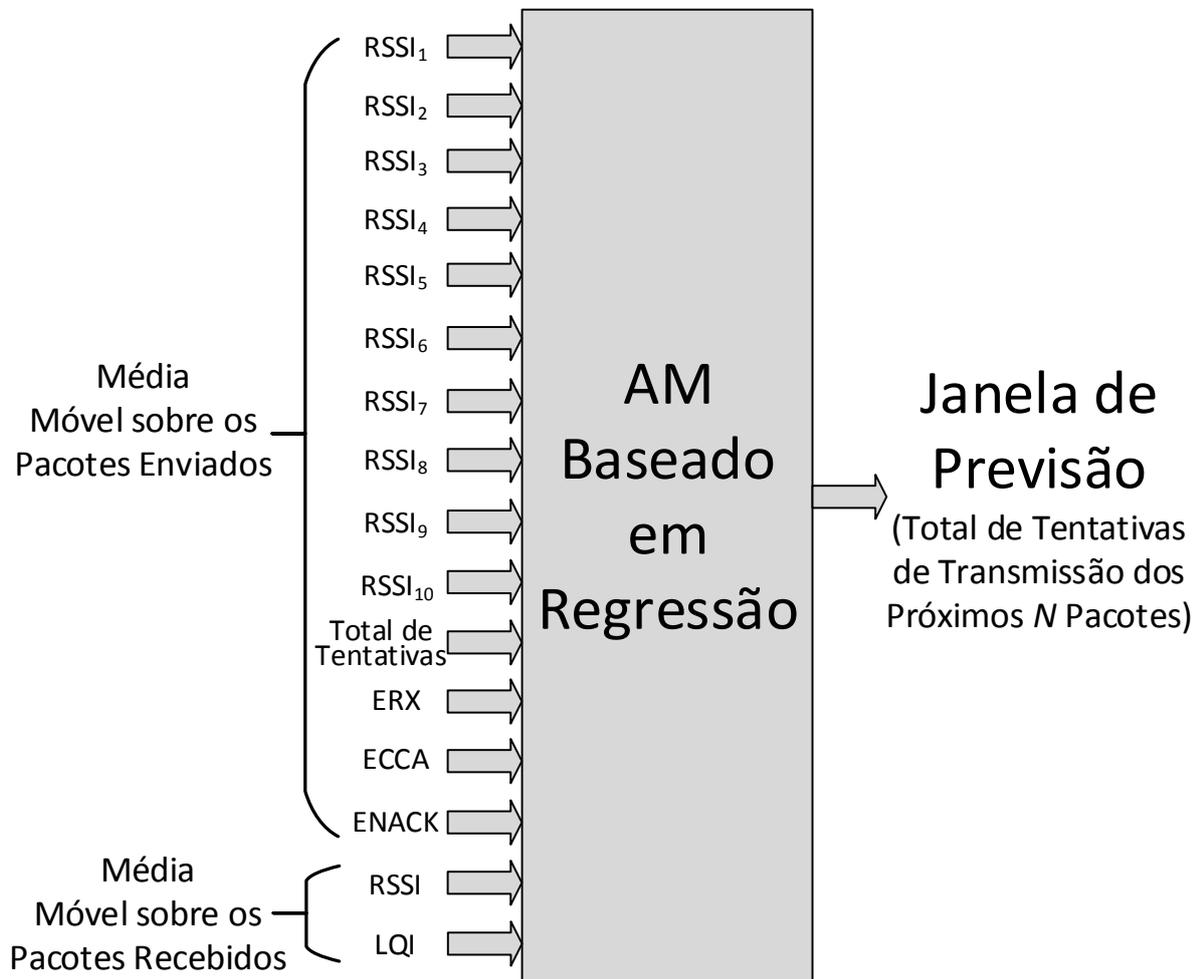


Figura 4.2. Modelo de predição: versão detalhada.

4.4 Algoritmo de Decisão de Espectro Proposto

Para o algoritmo proposto, são empregados dois tipos de nós sensores. O primeiro deles, denominado *Nó Central*, executa o algoritmo de Decisão de Espectro, o qual possui um preditor implementado. O outro nó, denominado *Nó Auxiliar*, por sua vez, se mantém fixo em um dos canais analisados. A função deste nó é se comunicar com o *Nó Central* quando necessário.

Apesar de empregarmos um *Nó Auxiliar* por canal analisado, o algoritmo pode ser modificado para que seja empregado somente um *Nó Auxiliar* para todos os canais

sensorizados, de forma a reduzir o consumo de recursos necessário para a Decisão de Espectro. A seguir é apresentada uma descrição passo-a-passo do funcionamento da solução de Decisão de Espectro, além de uma representação gráfica desse funcionamento na Figura 4.3.

1. Para cada canal:
 - a) O *Nó Central* envia pacotes de sondagem durante T segundos ao *Nó Auxiliar*, de forma a coletar dados referentes à transmissão;
 - b) O *Nó Auxiliar* envia pacotes de sondagem ao *Nó Central* durante T segundos, para que o *Nó Central* colete dados referentes à recepção.
 - c) É realizada a previsão para o respectivo canal, empregando os dados coletados;
2. O *Nó Central* seleciona o canal com melhor previsão, ou seja, com o menor valor de saída obtido;
3. O *Nó Central* transmite N pacotes de dados ao *Nó Auxiliar* localizado no canal selecionado, encaminhando o sumário do desempenho obtido ao computador, para mensurar o desempenho do algoritmo de troca de canal. É importante observar que este passo não tem influência direta no processo de decisão;
4. Reinicia-se o processo de decisão, retornando ao Passo 1.

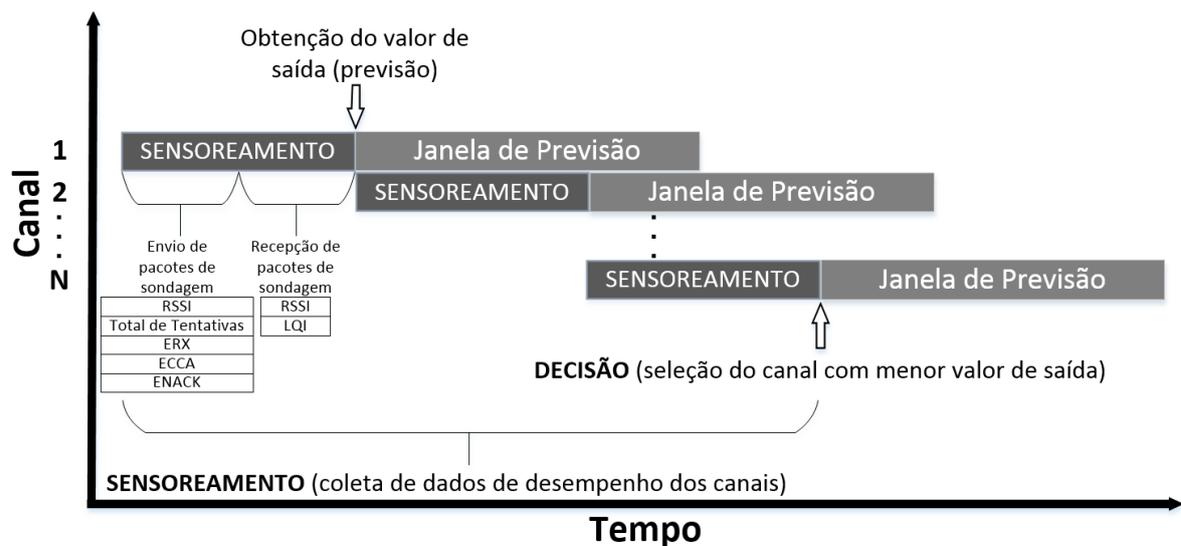


Figura 4.3. Algoritmo de Decisão de Espectro.

4.5 Configuração dos Experimentos de Coleta de Dados

Nesta Seção é descrita a configuração utilizada para a coleta dos dados a serem utilizados na construção das bases de treinamento, apresentando a plataforma utilizada, cenários de coleta considerados, além de outros parâmetros de medição da RSSF considerados relevantes.

4.5.1 Plataforma Utilizada

Foi escolhida a Plataforma Iris [Crossbow, 2008a] para coleta dos dados de desempenho a cada transmissão. A transmissão dos dados entre os nós da RSSF é feita por meio do rádio RF230, contido na plataforma Iris. A Tabela 4.1 apresenta as especificações técnicas do mesmo.

Tabela 4.1. Especificações técnicas para o rádio RF230.

Especificação		Observações
Faixa de Frequência	2.405 MHz a 2.480 MHz	Banda ISM, programável a passos de 1 MHz
Taxa de Transmissão	250 Kbps	Codificação Manchester
Potência de Transmissão	De -17 até +3 dBm	
Sensibilidade de Recepção	-101dBm	
Alcance em Ambiente Externo	>300m	1/4 onda, LOS (<i>Line Of Sight</i>)
Consumo de Corrente	17 mA	Modo de Transmissão
	16 mA	Modo de Recepção
	0,1 μ A	Modo <i>Sleep</i>

Para a programação dos nós sensores, foi utilizada a base MIB520, a qual também possui a função de receber os dados coletados via rádio e encaminhá-los ao computador.

A Figura 4.4 mostra um nó da plataforma Iris, e a base MIB520 por sua vez é mostrada na Figura 4.5.

4.5.2 Topologia da RSSF para Coleta dos Dados

Nos experimentos de coleta realizados, a RSSF projetada foi composta por 10 nós sensores Iris, cada um conectado à base MIB520, que por sua vez foi conectada ao



Figura 4.4. Nó sensor Iris [Crossbow, 2008a].



Figura 4.5. MIB520 [Open Automation, 2013].

computador por uma porta USB. A Figura 4.6 mostra a RSSF projetada, juntamente com dois roteadores utilizados para a geração de interferência, a qual será descrita na Seção 4.5.3 a seguir.

4.5.3 Cenários de Coleta Analisados

Os cenários de coleta foram escolhidos de forma a representar redes com variados graus de interferência. Para tanto, empregamos roteadores Wi-Fi para geração de interferência.

Os dados foram coletados em três cenários distintos, os quais são descritos com mais detalhes abaixo, juntamente com uma descrição de como a geração de interferência



Figura 4.6. RSSF projetada para coleta dos dados de desempenho.

pelos roteadores foi configurada.

1. **Baixa Interferência:** Cenário em que apenas os nós sensores transmitiam dados entre si, com um nível de interferência externa quase nulo;
2. **Média Interferência:** Além da transmissão de dados entre os nós da RSSF, foram também colocados dois roteadores transmitindo dados entre si, a baixa potência e distantes um do outro, na mesma faixa de frequência da RSSF;
3. **Alta Interferência:** Semelhante ao cenário de Média Interferência, os mesmos roteadores foram colocados próximos um do outro e da RSSF, transmitindo na mesma faixa de frequência e a alta potência.

4.5.4 Parâmetros de Transmissão

Foi definido um intervalo de transmissão de pacotes de 500 ms , com tamanho fixo de 28 bytes . Os dados de desempenho eram encaminhados ao computador a cada transmissão

finalizada. Foram coletadas 10 amostras de *RSSI* antes de cada transmissão. Por fim, os nós sensores foram colocados par-a-par, com o objetivo de reduzir os efeitos da aleatoriedade no desempenho dos algoritmos de AM. Foram coletados dados de desempenho em um intervalo de quatro horas.

4.6 Conclusão

Neste capítulo, foi apresentada a modelagem proposta para o problema, do ponto de vista teórico. Foi feita uma descrição do algoritmo de Decisão de Espectro proposto, no qual será implementado um dos modelos de predição gerados pelos algoritmos de AM. Do ponto de vista prático, foram apresentados todos os passos de coleta dos dados para a construção das bases de treinamento, a serem utilizadas na avaliação dos algoritmos de AM na ferramenta de análise.

A criação de um modelo de predição que busca aproximar o desempenho futuro da rede, tomando como base o comportamento da *RSSF* no passado sob a forma de média móvel, se mostrou adequada para implementação em nós sensores reais. Isso se deve principalmente ao fato de que os atributos coletados são em número reduzido e de fácil obtenção, o que contribui para a redução dos custos de processamento e memória.

Além disso, o uso de média móvel ponderada nos atributos de entrada permite reduzir a alta variabilidade observada no meio. A opção por preditores baseados em regressão se alia a um dos objetivos deste trabalho, que é obter um modelo de predição que realize uma aproximação do comportamento real dos canais, fato que é facilmente obtido por meio da definição de um intervalo numérico.

Capítulo 5

Análise dos Algoritmos de Aprendizizado de Máquina

Neste capítulo é apresentada uma análise detalhada da acurácia dos algoritmos de AM, segundo a modelagem adotada e descrita no Capítulo 4. Essas etapas precedem a implementação dos melhores modelos de predição em plataformas comerciais de RSSFs, que será descrita com mais detalhes no Capítulo 6.

As seções neste capítulo estão organizadas da seguinte forma. A Seção 5.1 mostra a metodologia de execução e avaliação dos algoritmos de AM, bem como quais algoritmos foram executados. A modelagem dos dados para as bases de treinamento teve seus parâmetros ajustados e otimizados na Seção 5.2 sob diferentes aspectos. Em seguida, a Seção 5.3 apresenta um ajuste do tempo de amostragem necessário para cada canal, ajuste esse necessário para a implementação dos melhores modelos de predição obtidos em nós sensores reais. Finalmente, a Seção 5.4 faz uma análise do impacto dos atributos de entrada sobre o desempenho dos algoritmos de AM. A Seção 5.5 fecha o capítulo com as conclusões.

5.1 Metodologia de Avaliação dos Algoritmos de AM

Para avaliar o desempenho dos algoritmos, foi utilizado o método *10-Fold Cross Validation*. Nele, o conjunto de instâncias da base de treinamento é dividido em dez partes, obtendo-se dez combinações Treinamento-Teste (nove partes para Treinamento, e uma para Teste). Cada execução realiza a divisão das partes de Teste de forma diferente, gerando assim diferentes combinações de dados. Cada algoritmo foi executado dez ve-

zes, portanto, foi obtido o desempenho em cem combinações Treinamento-Teste (dez para cada execução).

Foram consideradas duas métricas, que são apresentadas abaixo.

1. **Coefficiente de Correlação:** Quantifica o nível de aproximação dos valores gerados pelo modelo de predição com os valores reais da base de treinamento. Esse nível é definido por um valor entre 0 (correlação nula) e 1 (correlação máxima);
2. **Raiz do Erro Médio Quadrático:** O *Erro Médio Quadrático* representa a média das diferenças entre o valor real (contido na base de treinamento) e o valor previsto (gerado pelo modelo de predição), elevadas ao quadrado.

5.1.1 Algoritmos de AM Testados

Foi utilizada a ferramenta Weka para aplicação dos algoritmos de AM selecionados [Hall et al., 2009]. Foram aplicados dezesseis algoritmos, os quais foram selecionados seguindo dois critérios distintos. O primeiro deles foi o suporte do algoritmo de AM a atributos de entrada e saída do tipo numérico, como visto na descrição dos atributos (Seção 4.3).

O segundo critério foi o tempo de execução de cada algoritmo. Para esse último, foram feitas execuções preliminares para cada algoritmo que atendia ao primeiro critério, de forma a estimar o tempo de execução. Foram selecionados os algoritmos com tempo de execução inferior a uma semana, para fins de obtenção de resultados em tempo hábil.

Em razão do segundo critério, um dos algoritmos desconsiderados foi o *SVM* (*Support Vector Machine*), ainda que este seja atualmente o estado da arte entre os algoritmos de AM. Os algoritmos selecionados, os quais foram mantidos com a configuração padrão fornecida pela ferramenta Weka, são classificados e listados abaixo.

- Algoritmos baseados em função:

1. *Isotonic Regression*
2. *Linear Regression*
3. *MLP Regressor*
4. *Multilayer Perceptron*
5. *Pace Regression*
6. *RBF Regressor*

- Algoritmos baseados em busca:
 1. *KNN - K-Nearest Neighbours*
- Algoritmos de meta-aprendizado:
 1. *Additive Regression* – com *Decision Stump* como algoritmo base
 2. *Bagging* – com *REP Tree* como algoritmo base
- Algoritmos baseados em regra de decisão:
 1. *Conjunctive Rule*
 2. *Decision Table*
 3. *M5 Rules*
- Algoritmos baseados em árvore de decisão:
 1. *Decision Stump*
 2. *Extra Tree*
 3. *M5P*
 4. *REP Tree*

5.2 Testes Realizados

Os testes foram divididos em uma série de sub-etapas, as quais são descritas com mais detalhes nas seções a seguir, bem como os resultados obtidos, de forma a otimizar os parâmetros relacionados à modelagem de dados adotada. Cada sub-etapa procurou aperfeiçoar a anterior, com base nos melhores valores obtidos em cada uma. Aqui, propõe-se juntar as medições realizadas nos três cenários (descritos na Seção 4.5.3) em uma única base de dados, criando assim um único modelo de predição, que seja mais genérico e apresente menor consumo de recursos. Todos os valores apresentados nos resultados foram arredondados para duas casas decimais, para fins de simplificação.

5.2.1 Medições de Um Nó/Peso Variável

Esta sub-etapa teve como objetivo verificar a influência do valor de α sobre o desempenho dos algoritmos analisados. Foi utilizada uma hora de medições feita por um dos dez nós sensores, o qual foi escolhido de forma aleatória. Foram testados valores de α

entre 0,1 e 0,9, com passo de 0,2 entre duas bases de dados consecutivas, e o tamanho da janela de previsão foi fixado em 10.

As tabelas 5.1 e 5.2 apresentam o *Coefficiente de Correlação* e a *Raiz do Erro Médio Quadrático* obtidos para cada algoritmo e valores de α analisados. Vale ressaltar que os valores apresentados representam a média entre 10 execuções do mesmo algoritmo para uma determinada base, seguido de seu respectivo desvio padrão entre parênteses.

Para todas as métricas analisadas, o algoritmo *KNN* foi o que obteve o melhor desempenho. Entre os cinco pesos aqui considerados, o coeficiente de correlação máximo foi de 0,95, e o menor valor para a Raiz do Erro Médio Quadrático foi 5,94.

Foi também observado que a redução do valor de α causou um aumento no desempenho dos algoritmos de regressão em geral. Isso também já era esperado, uma vez que quanto menor o valor de α , menor será a variabilidade sobre os valores de entrada. Como aqui consideramos três cenários diferentes (Baixa, Média e Alta Interferência), uma menor variabilidade entre as entradas faz com que a base de treinamento se aproxime de três pontos distintos, cada um deles relacionado a um determinado cenário, o que simplifica as soluções geradas pelos algoritmos de regressão, consequentemente aumentando o seu desempenho. Considera-se aqui que essa aproximação seja um passo importante na criação de um modelo genérico de previsão de desempenho dos canais, já que um modelo cada vez mais simples reduz consideravelmente o tempo de previsão e de decisão.

5.2.2 Medições de Um Nó/Tamanho de Base Variável

Nesta sub-etapa, o objetivo principal foi verificar o impacto no desempenho dos algoritmos quando o intervalo de medição considerado era incrementado. Quanto maior o intervalo, maior o número de instâncias criadas na base de treinamento. Foram construídas bases com uma, duas, três e quatro horas de coleta de dados. O valor de α foi fixado em 0,1, uma vez que esse foi o peso que ofereceu o melhor desempenho na sub-etapa anterior, e o tamanho da janela de previsão foi novamente fixado em 10.

As tabelas 5.3 e 5.4 apresentam o valor médio e o desvio padrão obtidos com a execução de cada um dos algoritmos analisados, para as métricas *Coefficiente de Correlação* e *Raiz do Erro Médio Quadrático* respectivamente.

Não foram observados padrões de comportamento ou grandes variações para a métrica *Coefficiente de Correlação* cujos valores superaram 0,85 em todas as bases aplicadas. Para a métrica *Raiz do Erro Médio Quadrático*, foi observado que quando o intervalo considerado aumenta de uma para duas e três horas, o erro aumenta para

Tabela 5.1. Coeficiente de correlação: medições de um nó/peso variável.

Algoritmo Aplicado	Peso (α)				
	0,1	0,3	0,5	0,7	0,9
Isotonic Regression	0,91 (0,00)	0,88 (0,01)	0,83 (0,01)	0,82 (0,01)	0,82 (0,01)
Linear Regression	0,88 (0,01)	0,81 (0,01)	0,75 (0,01)	0,71 (0,01)	0,66 (0,01)
MLP Regressor	0,91 (0,00)	0,90 (0,01)	0,87 (0,01)	0,84 (0,01)	0,82 (0,02)
Multilayer Perceptron	0,91 (0,00)	0,90 (0,01)	0,88 (0,01)	0,85 (0,02)	0,83 (0,01)
Pace Regression	0,88 (0,01)	0,81 (0,01)	0,76 (0,01)	0,71 (0,01)	0,66 (0,01)
RBF Regressor	0,91 (0,00)	0,88 (0,01)	0,84 (0,01)	0,81 (0,01)	0,75 (0,04)
KNN	0,95 (0,00)	0,84 (0,01)	0,80 (0,01)	0,79 (0,01)	0,80 (0,01)
Additive Regression	0,91 (0,00)	0,88 (0,01)	0,86 (0,01)	0,85 (0,01)	0,86 (0,01)
Bagging REP Tree	0,92 (0,00)	0,91 (0,00)	0,90 (0,00)	0,90 (0,00)	0,90 (0,00)
Conjunctive Rule	0,91 (0,00)	0,87 (0,01)	0,81 (0,01)	0,80 (0,01)	0,82 (0,02)
Decision Table	0,91 (0,00)	0,89 (0,01)	0,84 (0,01)	0,82 (0,01)	0,82 (0,01)
M5 Rules	0,91 (0,01)	0,91 (0,00)	0,90 (0,01)	0,89 (0,01)	0,89 (0,01)
Decision Stump	0,91 (0,00)	0,86 (0,01)	0,80 (0,01)	0,80 (0,01)	0,79 (0,01)
Extra Tree	0,87 (0,01)	0,83 (0,01)	0,82 (0,01)	0,81 (0,01)	0,82 (0,01)
M5P	0,92 (0,00)	0,91 (0,00)	0,90 (0,01)	0,89 (0,01)	0,89 (0,01)
REP Tree	0,91 (0,00)	0,90 (0,01)	0,89 (0,01)	0,89 (0,01)	0,90 (0,01)

Tabela 5.2. Raiz do erro médio quadrático: medições de um nó/peso variável.

Algoritmo Aplicado	Peso (α)				
	0,1	0,3	0,5	0,7	0,9
Isotonic Regression	7,44 (0,19)	8,67 (0,24)	10,03 (0,25)	10,31 (0,25)	10,35 (0,25)
Linear Regression	8,73 (0,22)	10,65 (0,25)	11,87 (0,26)	12,74 (0,26)	13,52 (0,27)
MLP Regressor	7,37 (0,20)	7,94 (0,23)	8,87 (0,32)	9,76 (0,37)	10,31 (0,49)
Multilayer Perceptron	8,06 (1,02)	8,89 (1,29)	9,87 (1,58)	10,85 (2,03)	11,30 (1,78)
Pace Regression	8,69 (0,22)	10,60 (0,25)	11,82 (0,26)	12,74 (0,26)	13,52 (0,27)
RBF Regressor	7,65 (0,19)	8,55 (0,22)	9,78 (0,25)	10,66 (0,24)	11,93 (0,87)
KNN	5,94 (0,28)	10,35 (0,27)	11,32 (0,26)	11,65 (0,31)	11,44 (0,34)
Additive Regression	7,42 (0,19)	8,57 (0,23)	9,26 (0,21)	9,43 (0,20)	9,14 (0,22)
Bagging REP Tree	7,12 (0,19)	7,62 (0,19)	7,86 (0,20)	7,94 (0,19)	7,80 (0,21)
Conjunctive Rule	7,61 (0,19)	8,94 (0,26)	10,61 (0,31)	10,80 (0,34)	10,36 (0,54)
Decision Table	7,46 (0,20)	8,40 (0,23)	9,93 (0,27)	10,22 (0,28)	10,25 (0,30)
M5 Rules	7,45 (0,30)	7,66 (0,22)	8,02 (0,29)	8,30 (0,29)	8,07 (0,26)
Decision Stump	7,61 (0,19)	9,15 (0,27)	10,78 (0,27)	10,92 (0,26)	11,09 (0,22)
Extra Tree	9,18 (0,29)	10,37 (0,28)	10,82 (0,29)	11,01 (0,37)	10,78 (0,32)
M5P	7,26 (0,19)	7,60 (0,21)	8,01 (0,27)	8,30 (0,47)	8,06 (0,27)
REP Tree	7,44 (0,22)	7,83 (0,24)	8,07 (0,23)	8,15 (0,22)	7,95 (0,22)

Tabela 5.3. Coeficiente de correlação: medições de um nó/tamanho de base variável.

Algoritmo Aplicado	Tamanho da Base (em horas de coleta)			
	Uma	Duas	Três	Quatro
Isotonic Regression	0,91 (0,00)	0,91 (0,00)	0,91 (0,00)	0,91 (0,00)
Linear Regression	0,88 (0,01)	0,87 (0,00)	0,87 (0,00)	0,87 (0,00)
MLP Regressor	0,92 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)
Multilayer Perceptron	0,92 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)
Pace Regression	0,88 (0,01)	0,88 (0,00)	0,87 (0,00)	0,87 (0,00)
RBF Regressor	0,91 (0,00)	0,91 (0,00)	0,91 (0,00)	0,91 (0,01)
KNN	0,93 (0,01)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)
Additive Regression	0,92 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)
Bagging – REP Tree	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)
Conjunctive Rule	0,91 (0,00)	0,91 (0,00)	0,90 (0,00)	0,90 (0,00)
Decision Table	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)	0,91 (0,00)
M5 Rules	0,91 (0,01)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)
Decision Stump	0,91 (0,00)	0,91 (0,00)	0,90 (0,00)	0,90 (0,00)
Extra Tree	0,87 (0,01)	0,87 (0,01)	0,86 (0,00)	0,86 (0,00)
M5P	0,92 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)
REP Tree	0,91 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)

grande parte dos algoritmos, ao passo que cai quando o intervalo aumenta de três para quatro horas.

Uma possível causa para esse comportamento seria um aumento da variabilidade entre os dados de entrada, à medida em que mais instâncias são adicionadas à base de treinamento, o que provavelmente contribuiu para o aumento do erro.

5.2.3 Medições de Um Nó/Tamanho da Janela de Previsão Variável

Procurou-se aqui verificar a influência do tamanho da janela de previsão no desempenho de cada algoritmo analisado. É importante ressaltar que a janela de previsão, segundo a modelagem adotada e descrita nas seções 4.2 e 4.3, é representada pela soma do número de tentativas dos pacotes subsequentes. Portanto, uma variação no tamanho

Tabela 5.4. Raiz do erro médio quadrático: medições de um nó/tamanho de base variável.

Algoritmo Aplicado	Tamanho da Base (em horas de coleta)			
	Uma	Duas	Três	Quatro
Isotonic Regression	6,91 (0,18)	6,99 (0,14)	7,16 (0,08)	7,11 (0,09)
Linear Regression	8,23 (0,22)	8,36 (0,17)	8,44 (0,11)	8,46 (0,09)
MLP Regressor	6,84 (0,19)	6,94 (0,14)	7,08 (0,10)	7,03 (0,08)
Multilayer Perceptron	7,57 (1,43)	7,63 (0,92)	7,84 (0,99)	7,77 (1,28)
Pace Regression	8,18 (0,22)	8,29 (0,17)	8,36 (0,11)	8,37 (0,09)
RBF Regressor	6,92 (0,18)	7,01 (0,14)	7,13 (0,11)	7,21 (0,27)
KNN	6,49 (0,24)	6,97 (0,19)	7,28 (0,14)	7,52 (0,13)
Additive Regression	6,86 (0,19)	6,94 (0,13)	7,09 (0,08)	7,04 (0,09)
Bagging – REP Tree	6,69 (0,19)	6,69 (0,13)	6,72 (0,08)	6,78 (0,08)
Conjunctive Rule	7,06 (0,18)	7,11 (0,13)	7,47 (0,08)	7,38 (0,08)
Decision Table	6,83 (0,18)	6,95 (0,14)	7,06 (0,08)	7,03 (0,09)
M5 Rules	6,92 (0,25)	6,91 (0,16)	7,00 (0,08)	6,98 (0,09)
Decision Stump	7,06 (0,18)	7,12 (0,13)	7,48 (0,08)	7,38 (0,08)
Extra Tree	8,70 (0,25)	8,88 (0,20)	9,13 (0,14)	9,19 (0,12)
M5P	6,78 (0,18)	6,87 (0,15)	6,98 (0,08)	6,97 (0,09)
REP Tree	6,93 (0,21)	6,93 (0,15)	6,99 (0,10)	7,02 (0,10)

da janela de previsão é equivalente a uma variação no *número de pacotes subsequentes* considerado.

Quanto maior o tamanho da janela de previsão, maiores serão os valores de saída de cada uma das instâncias geradas, fazendo com que a Raiz do Erro Médio Quadrático obtida com diferentes tamanhos de janela de previsão fiquem em escalas numéricas diferentes entre si, dificultando a comparação entre esses valores. Por isso, com o intuito de realizar essa comparação, nesta sub-etapa as saídas foram normalizadas para uma escala entre 10 e 100, equivalendo-se à escala do tamanho de janela igual a 10 utilizado nas sub-etapas anteriores. Por exemplo, se o tamanho da janela for cinco, a menor soma é 5 (todas as transmissões subsequentes realizadas com apenas uma tentativa cada), que quando normalizada passa a ser 10, e a maior soma é 50 (todas as transmissões subsequentes com o máximo de 10 tentativas), que com a normalização passa a ser igual a 100.

Foram testados os tamanhos de janela 1, 5, 10, 15 e 20, e o valor de α foi novamente fixado em 0,1. Pelo fato da base de treinamento com uma hora de intervalo ter apresentado o melhor desempenho, para a maior parte dos algoritmos na sub-etapa anterior, este mesmo intervalo foi também utilizado nesta sub-etapa.

As tabelas 5.5 e 5.6 mostram os resultados obtidos para cada um dos algoritmos, para as métricas *Coefficiente de Correlação* e *Raiz do Erro Médio Quadrático* respectivamente.

A principal característica observada, para todos os algoritmos analisados, foi um crescimento logarítmico do coeficiente de correlação, bem como um decrescimento logarítmico da raiz do erro médio quadrático com o aumento do tamanho da janela de previsão. Um dos algoritmos que apresentou esse comportamento de forma mais expressiva foi o *KNN*, o qual teve coeficiente de correlação 0,40 para tamanho de janela igual a 1, com um salto de 0,50 no valor quando esse tamanho subiu para 5. Já para um tamanho de janela igual a 10, esse incremento no coeficiente foi de 0,06, e para os tamanhos 15 e 20 esse aumento foi de apenas 0,02. Para a métrica *Raiz do Erro Médio Quadrático*, os decrescimentos foram de 20,22, 3,87, 1,36 e 0,63.

Esse decrescimento logarítmico no erro motivou a execução das sub-etapas seguintes, onde também foram testados tamanhos de janela maiores que 20. A descoberta de um tamanho de janela ideal é essencial para a otimização do desempenho dos algoritmos analisados, pois como o objetivo final é o uso dos modelos construídos no *ranking* dos canais, é necessário que a previsão realizada em cada canal seja válida durante um tempo considerável mas ao mesmo tempo plausível, de forma que os resultados possam ser confrontados e assim seja descoberto o melhor canal.

5.2.4 Medições de Todos os Nós/Peso Variável

Neste caso, como o número de instâncias criadas para uma hora de medições foi extenso, uma vez que o número de instâncias nesta sub-etapa seria multiplicado por um fator de 10 (RSSF composta como 10 nós), foi utilizada apenas meia hora de medições. O principal objetivo desta sub-etapa foi verificar o impacto sobre o desempenho dos algoritmos analisados quando as medições de vários nós eram misturadas em uma única base, assim, seria possível criar um modelo de predição que fosse ainda mais genérico, eliminando a variabilidade entre diversos pontos da RSSF. Além disso, procurou-se testar valores de α menores que 0,1, uma vez que, em todas as sub-etapas anteriores foi observado um aumento no desempenho à medida em que o valor de α decrescia. Aqui, foram testados valores de α entre 0,01 e 0,09, com passo 0,02 entre duas bases de treinamento consecutivas. Também foi testada base com α igual a 0,1, para fins de

Tabela 5.5. Coeficiente de correlação: medições de um nó/tamanho da janela de previsão variável.

Algoritmo Aplicado	Tamanho da Janela de Previsão				
	1	5	10	15	20
Isotonic Regression	0,64 (0,01)	0,85 (0,00)	0,92 (0,00)	0,95 (0,00)	0,96 (0,00)
Linear Regression	0,63 (0,02)	0,82 (0,01)	0,89 (0,01)	0,92 (0,00)	0,93 (0,00)
MLP Regressor	0,64 (0,01)	0,86 (0,01)	0,93 (0,01)	0,95 (0,00)	0,96 (0,01)
Multilayer Perceptron	0,57 (0,08)	0,86 (0,00)	0,93 (0,00)	0,96 (0,00)	0,97 (0,00)
Pace Regression	0,63 (0,01)	0,82 (0,01)	0,90 (0,01)	0,92 (0,00)	0,93 (0,00)
RBF Regressor	0,64 (0,01)	0,86 (0,00)	0,93 (0,00)	0,95 (0,00)	0,96 (0,00)
KNN	0,40 (0,02)	0,90 (0,01)	0,96 (0,00)	0,98 (0,00)	0,98 (0,00)
Additive Regression	0,64 (0,01)	0,86 (0,00)	0,93 (0,00)	0,95 (0,00)	0,96 (0,00)
Bagging REP Tree	0,63 (0,02)	0,87 (0,00)	0,94 (0,00)	0,96 (0,00)	0,97 (0,00)
Conjunctive Rule	0,63 (0,02)	0,84 (0,00)	0,91 (0,00)	0,94 (0,00)	0,95 (0,00)
Decision Table	0,64 (0,01)	0,86 (0,00)	0,93 (0,00)	0,95 (0,00)	0,96 (0,00)
M5 Rules	0,64 (0,01)	0,87 (0,00)	0,93 (0,01)	0,96 (0,00)	0,97 (0,00)
Decision Stump	0,63 (0,02)	0,84 (0,00)	0,91 (0,00)	0,94 (0,00)	0,95 (0,00)
Extra Tree	0,43 (0,03)	0,80 (0,01)	0,90 (0,01)	0,94 (0,00)	0,95 (0,00)
M5P	0,64 (0,01)	0,87 (0,00)	0,94 (0,00)	0,96 (0,00)	0,97 (0,00)
REP Tree	0,64 (0,01)	0,86 (0,00)	0,93 (0,00)	0,96 (0,00)	0,97 (0,00)

Tabela 5.6. Raiz do erro médio quadrático: medições de um nó/tamanho da janela de previsão variável.

Algoritmo Aplicado	Tamanho da Janela de Previsão				
	1	5	10	15	20
Isotonic Regression	20,93 (0,34)	10,66 (0,28)	7,32 (0,25)	5,98 (0,20)	5,31 (0,13)
Linear Regression	21,20 (0,34)	11,63 (0,32)	8,44 (0,39)	7,16 (0,26)	6,61 (0,20)
MLP Regressor	20,83 (0,32)	10,52 (0,32)	7,04 (0,26)	5,59 (0,22)	5,01 (0,38)
Multilayer Perceptron	25,15 (3,36)	11,04 (0,94)	7,14 (0,39)	6,26 (1,68)	5,10 (0,70)
Pace Regression	21,21 (0,34)	11,61 (0,32)	8,39 (0,37)	7,10 (0,26)	6,55 (0,21)
RBF Regressor	20,83 (0,32)	10,32 (0,30)	7,06 (0,19)	5,71 (0,13)	5,01 (0,15)
KNN	29,31 (0,54)	9,09 (0,50)	5,22 (0,31)	3,86 (0,15)	3,23 (0,17)
Additive Regression	20,82 (0,33)	10,35 (0,27)	6,95 (0,19)	5,54 (0,13)	4,83 (0,16)
Bagging REP Tree	21,02 (0,35)	9,98 (0,27)	6,41 (0,19)	4,97 (0,10)	4,23 (0,18)
Conjunctive Rule	21,07 (0,34)	10,90 (0,25)	7,72 (0,19)	6,49 (0,14)	5,89 (0,11)
Decision Table	20,90 (0,37)	10,39 (0,26)	7,00 (0,18)	5,59 (0,15)	4,88 (0,17)
M5 Rules	20,88 (0,33)	10,16 (0,27)	6,77 (0,25)	5,34 (0,11)	4,59 (0,18)
Decision Stump	21,07 (0,34)	10,90 (0,25)	7,72 (0,19)	6,49 (0,14)	5,89 (0,11)
Extra Tree	28,48 (0,57)	12,89 (0,30)	8,32 (0,28)	6,57 (0,13)	5,64 (0,23)
M5P	20,86 (0,33)	10,17 (0,26)	6,61 (0,18)	5,26 (0,11)	4,50 (0,20)
REP Tree	20,84 (0,35)	10,39 (0,21)	6,77 (0,16)	5,29 (0,11)	4,45 (0,22)

comparação com os resultados obtidos na Seção 5.2.1, onde as bases possuíam dados provenientes de apenas um nó, e também foi testado esse mesmo valor. O tamanho da janela de previsão foi novamente fixado em 10.

As tabelas 5.7 e 5.8 mostram os resultados obtidos para cada um dos algoritmos, para as métricas *Coefficiente de Correlação* e *Raiz do Erro Médio Quadrático* respectivamente.

Quando comparados os resultados obtidos para a base com valor de α igual a 0,1 desta sub-etapa com os da Seção 5.2.1, onde foram utilizados dados provenientes de apenas um nó, fica evidente um aumento considerável no desempenho, para todas as métricas analisadas. Para a métrica *Coefficiente de Correlação*, a média de crescimento nos valores entre os 16 algoritmos foi de 0,02, valendo lembrar que todos os valores se apresentavam superiores a 0,8, para as medições de um nó apenas. Já para a métrica *Raiz do Erro Médio Quadrático*, também houveram quedas em todos os algoritmos, com média de queda em 0,18.

Outro fator que contribui com o aumento do Coeficiente de Correlação e a redução na Raiz do Erro Médio Quadrático é a redução do intervalo mensurado, que aqui foi reduzido para meia hora, ao contrário de uma hora como havia sido feito em sub-etapas anteriores. Ainda que, para uma base com intervalo menor e medições de todos os nós, o número de instâncias de treino tenha sido bem maior do que uma base com intervalo maior e medições de apenas um nó, percebe-se aqui também uma influência do intervalo considerado sobre o desempenho dos algoritmos de AM. Tal fato já havia sido percebido na sub-etapa da Seção 5.2.2, onde o intervalo mensurado foi variado. Portanto, no contexto de RSSFs, espera-se que a criação de um modelo simplificado de uso do canal tome como base informações coletadas em pequenos intervalos e de forma igualitária por todos os nós da Rede.

Com relação às bases de treinamento com valores de α menores que 0,1, não foram observadas variações consideráveis no Coeficiente de Correlação. Já para a métrica *Raiz do Erro Médio Quadrático*, para alguns dos algoritmos analisados percebeu-se um aumento no erro com a redução do valor de α de 0,03 para 0,01, o que indica um possível valor ideal para este parâmetro. Sob um ponto de vista prático, um valor de α ainda menor pode não trazer melhorias, uma vez que a atualização de informações sobre cada canal tende a ser ainda mais lenta, exigindo um tempo de sensoreamento mais elevado. O tempo de sensoreamento em razão do valor de α adotado será tratado nas próximas seções.

Tabela 5.7. Coeficiente de correlação: medições de todos os nós/peso variável.

Algoritmo Aplicado	Peso (α)					
	0,01	0,03	0,05	0,07	0,09	0,1
Isotonic Regression	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)
LinearRegression	0,93 (0,00)	0,93 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)	0,91 (0,00)
MLP Regressor	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)
Multilayer Perceptron	0,94 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)
Pace Regression	0,93 (0,00)	0,93 (0,00)	0,92 (0,00)	0,91 (0,00)	0,91 (0,00)	0,91 (0,00)
RBF Regressor	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)
KNN	0,99 (0,00)	0,99 (0,00)	0,98 (0,00)	0,98 (0,00)	0,96 (0,00)	0,95 (0,00)
Additive Regression	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)
Bagging – REP Tree	0,96 (0,00)	0,95 (0,00)	0,95 (0,00)	0,94 (0,00)	0,94 (0,00)	0,94 (0,00)
Conjunctive Rule	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)
Decision Table	0,94 (0,00)	0,94 (0,00)	0,94 (0,00)	0,93 (0,00)	0,93 (0,00)	0,93 (0,00)
M5 Rules	0,93 (0,00)	0,93 (0,00)	0,93 (0,01)	0,93 (0,00)	0,94 (0,00)	0,94 (0,00)
Decision Stump	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)	0,92 (0,00)
Extra Tree	0,96 (0,00)	0,94 (0,00)	0,92 (0,00)	0,91 (0,00)	0,90 (0,00)	0,90 (0,00)
M5P	0,95 (0,02)	0,94 (0,00)	0,94 (0,00)	0,94 (0,00)	0,94 (0,00)	0,94 (0,00)
REP Tree	0,95 (0,00)	0,94 (0,00)	0,94 (0,00)	0,94 (0,00)	0,93 (0,00)	0,93 (0,00)

Tabela 5.8. Raiz do erro médio quadrático: medições de todos os nós/peso variável.

Algoritmo Aplicado	Peso (α)					
	0,01	0,03	0,05	0,07	0,09	0,1
Isotonic Regression	7,35 (0,10)	7,30 (0,10)	7,33 (0,09)	7,35 (0,09)	7,39 (0,09)	7,42 (0,09)
LinearRegression	7,51 (0,09)	7,55 (0,10)	7,79 (0,10)	8,06 (0,11)	8,33 (0,11)	8,47 (0,11)
MLP Regressor	7,06 (0,09)	7,07 (0,10)	7,10 (0,10)	7,13 (0,13)	7,16 (0,15)	7,15 (0,11)
Multilayer Perceptron	7,67 (0,77)	7,75 (0,83)	7,76 (0,73)	7,80 (0,75)	7,78 (0,76)	7,79 (0,76)
Pace Regression	7,51 (0,09)	7,55 (0,10)	7,78 (0,10)	8,04 (0,10)	8,30 (0,11)	8,44 (0,11)
RBF Regressor	7,25 (0,10)	7,15 (0,09)	7,17 (0,09)	7,22 (0,11)	7,27 (0,09)	7,31 (0,13)
KNN	3,22 (0,05)	3,31 (0,06)	3,59 (0,08)	4,40 (0,11)	5,57 (0,11)	6,12 (0,11)
Additive Regression	7,12 (0,08)	7,12 (0,08)	7,12 (0,08)	7,13 (0,08)	7,16 (0,09)	7,17 (0,09)
Bagging – REP Tree	5,56 (0,09)	6,19 (0,08)	6,49 (0,08)	6,67 (0,08)	6,78 (0,08)	6,82 (0,08)
Conjunctive Rule	7,80 (0,08)	7,78 (0,08)	7,78 (0,09)	7,77 (0,09)	7,77 (0,08)	7,78 (0,08)
Decision Table	6,84 (0,10)	6,99 (0,09)	7,05 (0,08)	7,09 (0,09)	7,10 (0,09)	7,12 (0,09)
M5 Rules	7,35 (0,23)	7,53 (0,10)	7,54 (0,27)	7,10 (0,16)	7,05 (0,09)	7,06 (0,10)
Decision Stump	7,87 (0,09)	7,81 (0,09)	7,79 (0,09)	7,78 (0,09)	7,79 (0,09)	7,79 (0,09)
Extra Tree	5,66 (0,12)	6,98 (0,14)	7,81 (0,13)	8,37 (0,13)	8,76 (0,12)	8,92 (0,12)
M5P	5,99 (0,91)	6,84 (0,18)	6,83 (0,09)	6,95 (0,09)	7,01 (0,09)	7,03 (0,09)
REP Tree	5,99 (0,11)	6,70 (0,09)	6,95 (0,10)	7,05 (0,09)	7,10 (0,09)	7,10 (0,09)

5.2.5 Medições de Todos os Nós/Tamanho da Janela de Previsão Variável

Esta última sub-etapa teve como objetivo principal verificar a influência do tamanho da janela de previsão, agora com medições de todos os nós. Novamente foi utilizada apenas meia hora de medições de cada nó, de forma a reduzir o número de instâncias e assim reduzir o tempo de execução de cada algoritmo.

Optou-se por utilizar valor de α igual a 0,01, uma vez que este apresentou os melhores resultados na sub-etapa anterior para a maioria dos algoritmos. O tamanho da janela foi variado entre 1 e 30, seguindo o mesmo critério de normalização das saídas adotado na Seção 5.2.3.

Nesta sub-etapa, optou-se por relacionar apenas um conjunto limitado de algoritmos, como forma de orientação às primeiras soluções práticas a serem implementadas nos nós sensores. Foi utilizado o Teste T-Pareado, o qual também é disponibilizado na ferramenta Weka, com o objetivo de encontrar os cinco melhores algoritmos, de acordo com cada uma das métricas analisadas.

Vale lembrar que aqui foram criadas trinta bases de treinamento, definidas com tamanhos de janela de previsão diferentes entre si, e sobre as quais foram feitas dez execuções de cada algoritmo. As figuras 5.1 e 5.2 apresentam o Coeficiente de Correlação e a Raiz do Erro Médio Quadrático, respectivamente, obtidos nos cinco algoritmos de melhor desempenho. Os resultados obtidos para os outros algoritmos foram omitidos em razão de limitações de representação gráfica, além do fato de que os algoritmos omitidos apresentaram desempenho semelhante aos aqui mostrados.

Para as duas métricas analisadas, observou-se um comportamento logarítmico em todos os algoritmos selecionados, com o aumento do tamanho da janela de previsão. Esse fator indica a existência de um possível padrão de comportamento do meio ao longo de um intervalo específico, aqui representado pela janela de previsão.

A detecção de um padrão de comportamento faz com que as métricas tenham convergência sobre um determinado máximo global (para a métrica *Coeficiente de Correlação*) ou mínimo global (para a métrica *Raiz do Erro Médio Quadrático*), os quais não podem ser melhorados devido a algum ruído proveniente do meio ou alguma variação do mesmo, impossível de ser representada em um modelo de regressão segundo a modelagem de dados adotada.

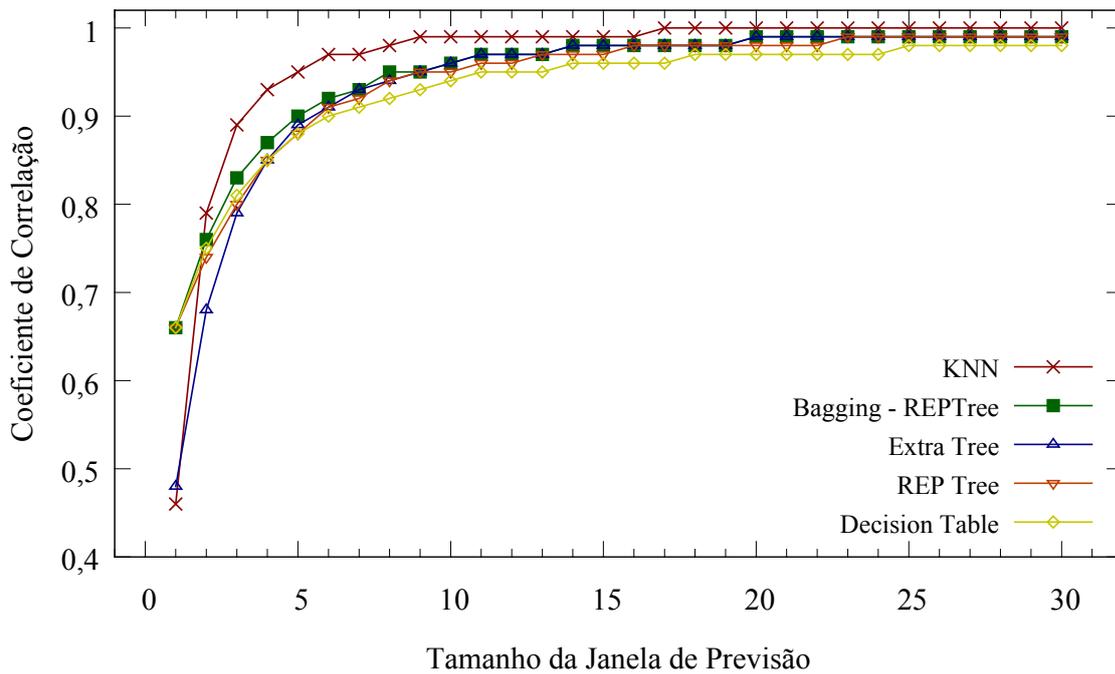


Figura 5.1. Coeficiente de correlação: medições de todos os nós/tamanho da janela de previsão variável.

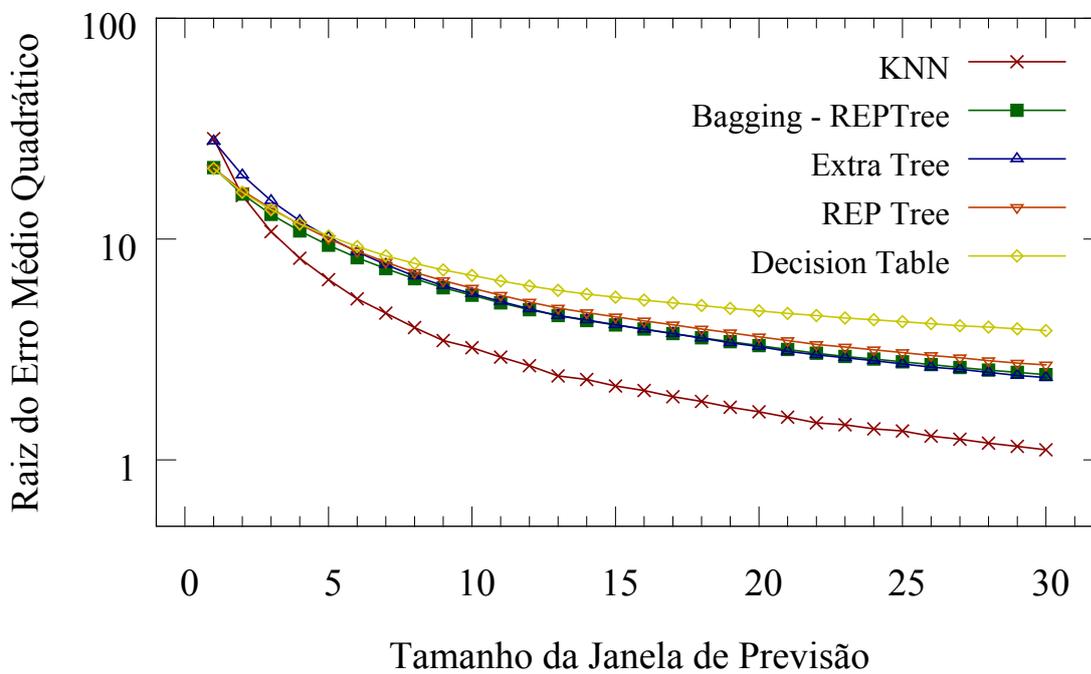


Figura 5.2. Raiz do erro médio quadrático: medições de todos os nós/tamanho da janela de previsão variável.

5.3 Ajuste dos Parâmetros de Sensoramento

Como visto nas seções anteriores, quanto menor o peso sobre valores de entrada do passado, definido no modelo pelo parâmetro α , melhor o desempenho sobre os algoritmos de regressão, uma vez que a variação de valor entre duas instâncias consecutivas é menor, e assim os valores da base se aproximam de uma função específica. Entretanto, a criação de um modelo por meio da utilização de um α pequeno pode trazer consequências práticas adversas, já que o tempo de convergência deverá ser maior. Em outras palavras, será necessário que o nó faça o sensoramento em cada canal durante um tempo maior, para que a previsão tenha maior convergência sobre as características atuais do meio e tenha maior durabilidade, permitindo que a mesma seja comparada com as previsões dos outros canais.

Partindo-se do pressuposto de que os valores se repitam indefinidamente, a Figura 5.3 mostra a taxa na qual os valores são atualizados, segundo a modelagem de média móvel sob diferentes valores de α . Foram testados valores de α iguais a 0,01, 0,02, 0,03 e 0,04. O eixo das ordenadas indica a porcentagem de aproximação na qual se encontra o valor da média móvel em relação ao valor total, e o eixo das abscissas indica o tempo decorrido, onde uma unidade é equivalente à geração de um novo valor de entrada.

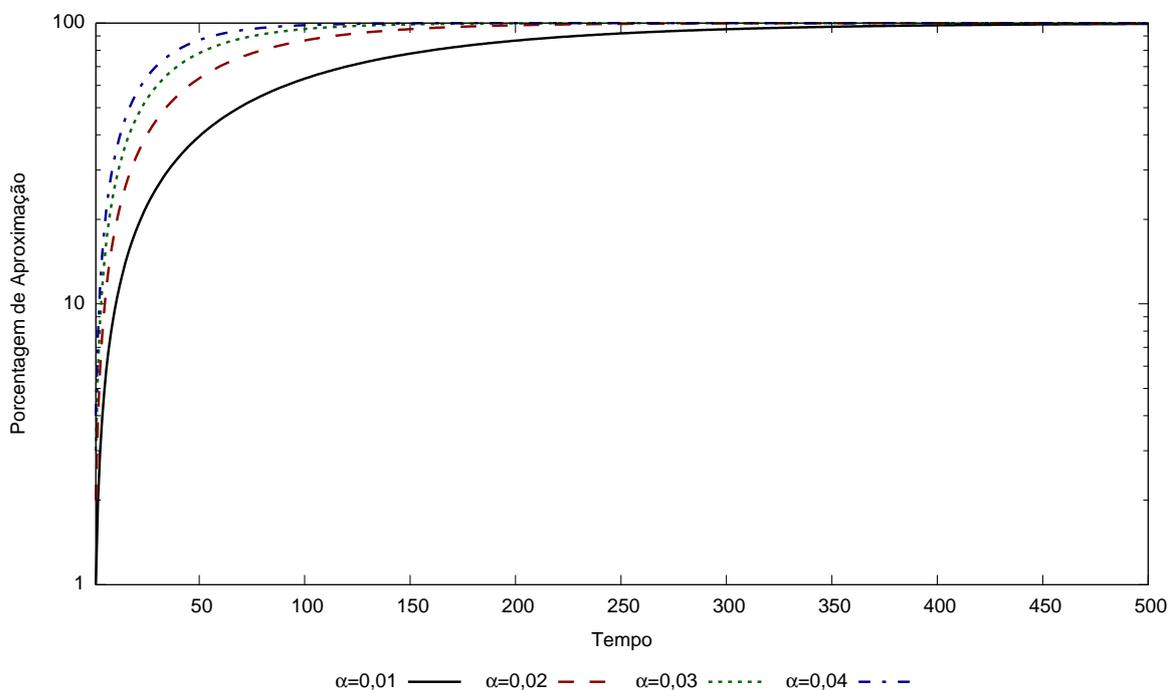


Figura 5.3. Atualização da média móvel/tempo decorrido.

Por meio da Figura, pode-se observar que há um crescimento exponencial no

tempo necessário para que a média móvel tenha 100% de aproximação do valor total, à medida em que o valor de α decresce, porcentagem essa que na prática é considerada aqui como uma aproximação total das condições atuais do meio. Por isso, o uso de uma base de treinamento com valor de α reduzido demandará em um tempo de sensoriamento por canal exponencialmente mais elevado, o que pode tornar inviável seu uso em um cenário real.

Portanto, é necessário que haja uma calibração da quantidade de pacotes enviados e recebidos para atualização dos atributos de entrada, o que irá determinar o tempo de sensoriamento necessário para cada canal. Assim, não são mais consideradas somente as métricas de desempenho dos algoritmos de AM, as quais são fornecidas pela ferramenta Weka, mas também a relação custo/benefício proporcionada pelo bom desempenho desses algoritmos, aqui inicialmente representada pelo tempo necessário para coleta dos dados de desempenho e seu conseqüente custo energético.

O ajuste do tempo de sensoriamento também terá influência direta no tamanho da janela de previsão, a qual deverá ser ajustada de acordo com o tempo de sensoriamento escolhido e o número de canais analisados. Ainda que o aumento nesse tamanho cause melhoras no desempenho dos algoritmos de aprendizagem, como visto em seções anteriores, é necessária a definição de um valor que seja plausível a nível prático, ou seja, que o tempo de sensoriamento tenha uma relação de proporcionalidade com o tamanho da janela de previsão.

Como visto na Equação 4.1, as instâncias de uma base de treinamento são construídas por meio da definição de uma média móvel, a qual é ajustada pelo fator α . Partindo-se do mesmo cenário da Figura 5.3, onde um valor se repete sucessivas vezes, podemos representar a sua convergência na forma de média móvel, a qual é apresentada a seguir:

$$Valor = Valor \times \alpha \times \sum_{i=0}^n (1 - \alpha)^i \quad (5.1)$$

Inicialmente o valor atual é ponderado com fator α . No intervalo seguinte, o valor resultante da multiplicação anterior é ponderado com fator $(1 - \alpha)$, uma vez que este agora faz parte do passado, e então é somado com o valor atual que é ponderado com fator α . Em seguida, o valor resultante dessa soma é mais uma vez multiplicado por $(1 - \alpha)$ e assim sucessivamente, fazendo com que a participação de valores mais antigos na soma final seja cada vez menor. Em outras palavras, os atributos de entrada a serem utilizados para a previsão de desempenho em um dado instante de tempo serão resultantes da soma de valores de desempenho obtidos com a transmissão e recepção de uma série de pacotes, cuja participação na soma é menor com o aumento da distância

temporal.

Temos então que, para um determinado valor de α , são necessários n termos no somatório da Equação 5.1 para que a soma seja equivalente a *Valor*, ainda com a premissa de que esse atributo se repita sucessivas vezes. A Equação 5.1, quando manipulada algebricamente se resume na seguinte expressão:

$$(1 - \alpha)^{n+1} = 0 \quad (5.2)$$

O valor à direita da Equação 5.2 representa o fator de distância da soma ponderada do valor real, o qual se encontra em um intervalo entre 0 (distância nula do valor real) e 1 (distância infinita/teórico). Como visto nesta Equação, em um cenário ideal o fator de distância deve ser nulo, o que em prática é inviável pois os termos da série tendem a reduzir indefinidamente, causando a necessidade de uma série com quantidade infinita de termos, que no caso em questão seria a transmissão ininterrupta de pacotes para coleta de dados de desempenho. Portanto, a Equação 5.2 é atualizada com a substituição do valor nulo à direita por um fator p , aqui denominado de *fator de distância* como mencionado inicialmente.

$$(1 - \alpha)^{n+1} = p \quad (5.3)$$

A Figura 5.4 apresenta a quantidade aproximada de pacotes enviados necessária para que os atributos de entrada se aproximem do valor real, obtida por meio da Equação 5.3. Foram considerados três fatores de distância distintos: 0,001 (0,1%), 0,01 (1%) e 0,1 (10%). Para cada fator, foi obtida a quantidade de pacotes necessária para cada valor de α testado em todos os testes da Seção 5.2.

O tempo de transmissão de um pacote é um dos atributos que permitem aproximar o tempo de sensoramento necessário para cada canal. É preciso que esse tempo seja estimado na prática para o pior caso, considerando-se todo o processo desde a leitura do RSSI do meio antes do início da primeira tentativa de transmissão, até a espera mal sucedida pelo pacote ACK na última tentativa realizada. No pior caso, na plataforma de testes selecionada, o nó realiza dez tentativas de transmissão com falha, das quais cinco são em razão do algoritmo CCA indicar que o canal está ocupado, e cinco em razão do pacote ACK não ter sido recebido. Esse caso acarreta no maior tempo de transmissão possível em todos os cenários analisados, pois acumula o tempo para execução do algoritmo CCA, o tempo de envio do pacote e a espera mal-sucedida pela confirmação.

Para que todos esses tempos fossem obtidos, foram colocados dois nós com transmissão unilateral de pacotes de 28 *bytes* em dois canais distintos durante um segundo

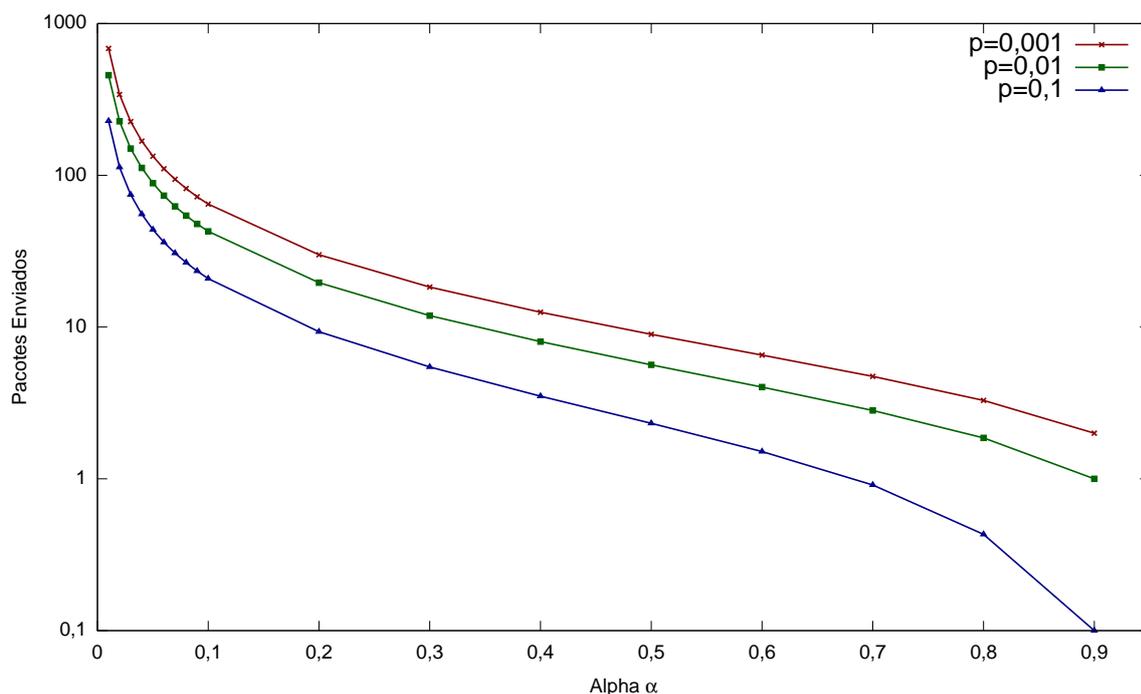


Figura 5.4. Ajuste do valor de α : quantidade de pacotes enviados.

cada. Foram realizadas 1.000 repetições do procedimento acima a intervalos de 2 segundos, sendo que em cada repetição obteve-se o tempo médio de medição do RSSI e o tempo médio de espera pelo pacote ACK entre os dois canais utilizados.

Como resultado, foi obtida uma média de 96,017 pacotes enviados por canal, com tempo médio de mensuração do RSSI de 1,264 milissegundos, e tempo médio de espera pelo pacote ACK de 0,673 milissegundos. A partir desses valores, obtém-se um tempo médio de envio por pacote de aproximadamente 8,35 milissegundos, desconsiderando-se o tempo de mensuração do RSSI, o tempo de espera pelo pacote ACK e o tempo execução do algoritmo CCA, o qual é realizado antes de cada tentativa de transmissão e é pré-definido em 128 microssegundos [Atmel, 2006].

É importante observar que estes valores são apenas estimativas do tempo máximo para cada etapa de transmissão de um pacote, uma vez que os tempos foram calculados em torno de outras funções executadas no nó sensor, inerentes ao sistema operacional TinyOS, o qual é o sistema padrão da plataforma utilizada. Além disso, o tempo de envio de um pacote mencionado anteriormente considera que os pacotes enviados em um segundo (96,017) tenham ocupado por completo esse intervalo, o que pode não acontecer em determinados momentos. Isso faz com que o tempo médio de envio aumente em relação ao tempo real.

É possível, portanto, estimar o tempo de sensoriamento de cada canal por meio

da equação 5.4 a seguir, considerando-se o pior caso citado anteriormente:

$$T_{sensing} = N_{pkt} * (T_{RSSI} + 5 * (2 * T_{CCA} + T_{send} + T_{ACKmax})) + ((N_{pkt} - 1) * T_{pktWait}) \quad (5.4)$$

onde

- $T_{sensing}$ é o tempo de sensoreamento resultante;
- N_{pkt} é o número de pacotes que precisam ser transmitidos em cada canal;
- T_{RSSI} é o tempo médio de medição do RSSI antes do início da transmissão;
- T_{CCA} é o tempo de espera pela execução do algoritmo CCA, o qual é pré-definido em 128 microssegundos, como dito anteriormente [Atmel, 2006].
- T_{send} é o tempo médio de envio do pacote, em uma tentativa;
- T_{ACKmax} é o tempo máximo de espera pelo pacote ACK. No TinyOS, o tempo de espera é definido em 0,8 milissegundos;
- $T_{pktWait}$ é o intervalo de transmissão entre pacotes distintos, o qual foi definido empiricamente em 500 milissegundos.

A Tabela 5.9 mostra o tempo necessário para envio em cada canal, em milissegundos, para cada um dos valores de α e fatores de distância considerados na Figura 5.4. É importante ressaltar que os tempos aqui apresentados são relacionados apenas ao envio de pacotes, por isso, o mesmo tempo também deve ser utilizado na recepção de pacotes, com o objetivo de atualizar os dados de desempenho relacionados a esse processo.

Como pode ser visto na Tabela 5.9, para alguns dos fatores de distância e valores de α analisados, é necessário um tempo de sensoreamento consideravelmente elevado, em razão da taxa de atualização da média móvel ser reduzida. Tais valores inviabilizam o seu uso em uma plataforma de RSSFs real, pois apesar de permitir a obtenção de previsões mais precisas, impede a transmissão de dados úteis por um tempo prolongado.

5.4 Teste de Sensibilidade dos Atributos de Entrada

Nesta seção é realizado um teste de sensibilidade sobre os atributos de entrada. O principal objetivo neste teste é descobrir o fator médio de contribuição de cada atributo

Tabela 5.9. Ajuste do tempo de sensoriamento por canal (em milissegundos).

α	Fator de Distância		
	0,001	0,01	0,1
0,01	375.803	250.186	124.569
0,02	186.425	123.934	61.443
0,03	123.298	81.849	40.401
0,04	91.732	60.805	29.879
0,05	72.792	48.178	23.565
0,06	60.163	39.760	19.355
0,07	51.142	33.745	16.349
0,08	44.375	29.234	14.093
0,09	39.111	25.725	12.338
0,1	34.899	22.917	10.934
0,2	15.925	10.267	4.610
0,3	9.571	6.031	2.491
0,4	6.366	3.895	1.423
0,5	4.416	2.595	773
0,6	3.085	1.707	330
0,7	2.097	1.049	0
0,8	1.305	520	0
0,9	597	48	0

sobre a previsão realizada. Em outras palavras, temos como principal objetivo descobrir a relevância de cada informação coletada sobre a análise da qualidade do canal, em termos do número de tentativas de transmissão dos pacotes subsequentes.

Por meio destes testes, é possível descobrir se existem atributos mais importantes que outros, sinalizando assim a possibilidade de uma coleta mais elaborada desses atributos. Ao mesmo tempo, é possível descobrir atributos que causam pouco ou nenhum impacto, ou até mesmo pioram a previsão, o que sugeriria a eliminação desse atributo, simplificando o modelo de predição.

É importante ressaltar que estes testes estão fora do escopo prático deste trabalho, ou seja, os resultados aqui apresentados não foram aplicados nas implementações em RSSF real, descritas no Capítulo 6, onde foram utilizados todos os atributos apresenta-

dos em capítulos anteriores. Com isso, esses resultados podem ser vistos apenas como uma base para os trabalhos futuros, fornecendo direções que permitam aperfeiçoar os modelos de predição implementados.

5.4.1 Metodologia dos Testes

Em suma, para obter o impacto de cada atributo sobre a previsão de cada algoritmo testado, foi criada uma série de bases de treinamento, todas com o mesmo conjunto de instâncias. Para cada base, retiramos um dos atributos de entrada. Como visto na Seção 4.3.1, temos dezesseis atributos, os quais são listados a seguir.

- $RSSI_1$ a $RSSI_{10}$: Média de cada uma das dez amostras de RSSI coletadas antes do início de cada transmissão;
- Tx_{Try} : Média do total de tentativas de transmissão;
- ERX : Média de falhas do tipo ERX;
- $ECCA$: Média de falhas do tipo ECCA;
- $ENACK$: Média de falhas do tipo ENACK;
- Rx_{RSSI} : RSSI médio dos últimos pacotes recebidos;
- Rx_{LQI} : LQI médio dos últimos pacotes recebidos.

Portanto, foram criadas dezesseis bases de treinamento, cada uma com um dos atributos suprimido, além da base com todos os atributos. Por meio das bases aqui criadas, foi possível obter as diferenças de desempenho, em termos do *Coefficiente de Correlação* e da *Raiz do Erro Médio Quadrático*, entre a base com todos os atributos, e cada uma das bases com um dos atributos suprimido.

Uma das bases de treinamento aqui criadas, a qual possuía todos os parâmetros, também foi a base utilizada em parte dos testes práticos, mais especificamente nos dois últimos testes (seções 6.4 e 6.5). Essa base possui diferenças em relação à base utilizada nos dois testes anteriores (seções 6.2 e 6.3), as quais serão descritas com mais detalhes ao longo do Capítulo 6.

Com base na Equação 5.2, escolhemos o valor de $\alpha = 0,7$ para a base de treinamento a ser utilizada, e com isso na construção das outras bases. Consideramos um fator de aproximação das condições reais do meio igual ou menor que 1% (fator p na Equação 5.2). Com isso, o tempo necessário para coleta dos atributos relativos ao envio de pacotes, em cada canal, é de aproximadamente um segundo. O tamanho

da janela de previsão foi definido empiricamente em aproximadamente seis minutos, tomando-se como base os resultados obtidos nas Seções 5.2.3 e 5.2.5 onde foi observado que, com o aumento do tamanho da janela de previsão, há um aumento no desempenho dos algoritmos de AM.

Através dos resultados obtidos na Seção 5.2.2, a qual apresentou testes com bases de treinamento de tamanhos diferentes, ou seja, com intervalos de medição diferentes, observou-se que o desempenho dos algoritmos de AM se mantinha elevado com tamanhos de base menores. Com base nessa premissa, com o intuito de se obter modelos de predição simplificados, foram utilizados apenas dois minutos e meio de medição de cada nó sensor, para cada cenário de coleta. Com isso, obteve-se uma base de treinamento para os testes de sensibilidade com cerca de 20.000 instâncias.

Todos os testes foram realizados na ferramenta Weka. Cada algoritmo de AM foi executado sob a mesma metodologia utilizada nos testes anteriores, a qual é apresentada na Seção 5.1, onde cada base aqui criada foi aplicada.

5.4.2 Resultados

As tabelas 5.10 e 5.11 apresentam os resultados obtidos, segundo as métricas *Coefficiente de Correlação* e *Raiz do Erro Médio Quadrático*, respectivamente.

Para cada algoritmo, é mostrada a variação média no desempenho, entre todas as execuções para cada remoção de atributo realizada, além do desempenho sobre a base com todos os atributos. Valores negativos representam queda em relação ao obtido com a base de treinamento completa, ao passo que valores positivos representam crescimento. Portanto, para a métrica *Coefficiente de Correlação*, valores positivos representam melhora no desempenho com a remoção do atributo, ao passo que para a métrica *Raiz do Erro Médio Quadrático* representam uma piora no desempenho.

Algoritmos cujo desempenho médio não sofreu variação com a retirada do atributo são assinalados com um traço apenas. Para os dez atributos representados pelas amostras de RSSI, é mostrado o intervalo de valores obtido.

Não houveram melhorias significativas, segundo a métrica *Coefficiente de Correlação*, uma vez que estes já se apresentavam superiores a 0,80, com exceção dos algoritmos *Linear Regression* e *Pace Regression*, que tiveram correlação média de 0,78 e 0,79 respectivamente para a base com todos os atributos. Entretanto, segundo a métrica *Raiz do Erro Médio Quadrático*, devido à maior granularidade desta, observou-se que a remoção de alguns atributos gerou melhoras no desempenho médio, em relação à base com todos os atributos. Essas melhorias se deram predominantemente com a remoção dos atributos relacionados ao RSSI coletado antes de cada transmissão, bem como o

Tabela 5.10. Teste de sensibilidade: variação no coeficiente de correlação.

Algoritmo	Remoções							Base Completa
	$RSSI_{1-10}$	Tx_{Try}	ERX	$ECCA$	$ENACK$	Rx_{RSSI}	Rx_{LQI}	
Isotonic Regression	-	-	-	-0,04	-	-	-	0,90
Linear Regression	-	-	-	-	0,01	-	-	0,78
MLP Regressor	-	-	-	-0,01	-0,01	-	-	0,90
Multilayer Perceptron	-	-0,01	-0,01	-0,01	-0,01	-	-	0,88
Pace Regression	-	-0,01	-0,01	-0,01	-	-	-	0,79
RBF Regressor	-	-	-	-	-	-	-	0,87
KNN	0,00 a 0,01	-0,01	-	-	-	-	-	0,88
Additive Regression	-	-	-	-0,03	-0,02	-	-	0,92
Bagging REP Tree	-	-	-	-0,01	-0,01	-	-	0,98
Conjunctive Rule	-	-	-	-0,03	-	-	-	0,86
Decision Table	-	-0,02	-	-	-	-	-	0,83
M5 Rules	-	-	-	-0,01	-0,01	-	-	0,97
Decision Stump	-	-	-	-0,04	-	-	-	0,86
Extra Tree	-	-	-	-0,02	-0,01	-	-	0,95
M5P	-	-	-	-0,01	-0,01	-	-	0,97
REP Tree	-0,01 a 0,00	-	-0,01	-0,03	-0,02	-	-	0,98

Tabela 5.11. Teste de sensibilidade: variação na raiz do erro médio quadrático.

Algoritmo	Remoções							Base Completa
	$RSSI_{1-10}$	$TxTry$	ERX	$ECCA$	$ENACK$	$RxRSSI$	$RxLQI$	
Isotonic Regression	-	-	-	1,24	-	-	-	6,97
Linear Regression	-0,01 a 0,01	-	-0,07	-0,04	-0,08	-	-	10,18
MLP Regressor	-0,07 a -0,01	-	-0,01	0,31	0,25	-0,08	-0,08	7,09
Multilayer Perceptron	-0,12 a 0,09	-0,05	-0,02	0,07	0,04	-0,09	-0,09	9,15
Pace Regression	0,00 a 0,01	0,13	0,07	0,10	0,06	-	-	10,04
RBF Regressor	-0,03 a 0,05	0,09	-0,04	0,12	0,04	-0,09	-0,09	8,00
KNN	-0,24 a 0,05	0,34	0,11	0,15	0,03	-0,01	-0,01	7,82
Additive Regression	-0,02 a 0,00	0,01	0,02	1,16	0,67	-	-	6,38
Bagging REP Tree	-0,02 a 0,07	-0,01	0,12	0,87	0,74	-	-	3,09
Conjunctive Rule	-0,01 a 0,00	-	-	0,86	-0,01	-	-	8,22
Decision Table	-0,07 a 0,01	0,33	-	-	-	-0,06	-0,06	9,17
M5 Rules	-0,22 a -0,07	-0,12	0,04	0,69	0,78	-0,02	-0,02	3,90
Decision Stump	-	-	-	0,97	-	-	-	8,27
Extra Tree	-0,01 a 0,11	0,10	0,17	0,79	0,60	0,10	0,10	5,15
M5P	-0,14 a 0,05	-0,04	0,09	0,52	0,77	0,02	0,02	3,75
REP Tree	-0,01 a 0,05	-0,01	0,10	1,40	1,17	-	-	3,55

RSSI e o LQI dos últimos pacotes recebidos. Uma das causas para a melhora, com a remoção das amostras de RSSI seria a baixa resolução espacial dos valores coletados antes de cada transmissão, devido a limitações de *hardware* inerentes ao modelo de rádio utilizado na plataforma Iris, fazendo com que esses valores comportassem como ruído no modelo de predição.

Para ambas as métricas analisadas, houve uma queda no desempenho quando os atributos *ECNA* e *ENACK* foram removidos, valendo lembrar que estes representam na prática a média de ocorrências de canal ocupado e colisões, respectivamente. Com isso, verifica-se a importância de um estudo mais aprofundado sobre esses dois atributos, fornecendo-os aos algoritmos de AM sob modelagens diferentes, além das utilizadas neste trabalho, de forma a melhorar ainda mais o desempenho dos modelos de predição.

Observa-se também, de modo geral, que a remoção de cada um dos atributos causa melhoras ou pioras de magnitudes diferentes sobre o desempenho de cada algoritmo de AM. Isso indica a necessidade de uma análise prévia, atributo a atributo, como forma de determinar o melhor conjunto para os algoritmos de AM. A detecção de um atributo, cuja remoção melhora consideravelmente o desempenho, ou o piora em magnitude quase nula é aqui considerado um fator positivo, uma vez que simplifica o espaço de atributos de entrada. Além disso, esta detecção favorece a simplificação do modelo de predição, reduzindo conseqüentemente a demanda por memória e processamento.

Por fim, como visto nas tabelas 5.10 e 5.11, não foram observados atributos que quando retirados causavam melhora no desempenho de todos os algoritmos de AM. Isso mostra que, para o conjunto de atributos de entrada utilizado, não existem parâmetros desnecessários ou que introduzam ruído.

5.5 Conclusão

Neste capítulo, foi feita uma descrição de todos os passos realizados para a análise dos algoritmos de AM, segundo a modelagem adotada para as bases de treinamento.

A realização de testes de desempenho, ajustando-se os parâmetros inerentes à modelagem sob a forma de média móvel, bem como o tamanho da janela de previsão e o tempo de sensoramento por canal, permitiram obter uma direção sobre qual base de treinamento deveria ser construída para obtenção dos modelos de predição a serem implementados em nós sensores reais.

Por meio dos testes de sensibilidade, foi possível descobrir a importância de cada um dos atributos de entrada, criando direções para futuras melhorias na modelagem aqui adotada, no sentido de simplificar ainda mais os modelos de predição.

Capítulo 6

Avaliação dos Modelos de Predição em uma RSSF Real

Este capítulo apresenta a avaliação experimental dos modelos de predição obtidos pelos melhores algoritmos de AM, verificados na ferramenta de análise, de forma a obter sua eficácia em uma RSSF real em cenários de teste distintos. A Seção 6.1 faz uma descrição dos modelos de predição escolhidos, bem como dos *baselines* implementados. Em seguida, as seções de 6.2 a 6.5 apresentam os testes realizados, além dos resultados obtidos em cada um.

6.1 Descrição das Implementações

Para solução de Decisão de Espectro, foram implementados os modelos de predição obtidos pelos quatro melhores algoritmos de AM, os quais eram passíveis de implementação nos nós sensores (executável gerado inferior ao tamanho de ROM e RAM da plataforma *Iris*), segundo a métrica *Raiz do Erro Médio Quadrático*.

Uma vez que a ferramenta Weka é desenvolvida em linguagem Java, os algoritmos selecionados tiveram seu código-fonte modificado na ferramenta, de forma que fosse obtido como resultado textual os modelos de predição em linguagem nesC, a qual é a linguagem utilizada pelo sistema operacional TinyOS para o desenvolvimento dos aplicativos dos nós sensores. Os algoritmos de AM selecionados são listados abaixo:

1. *REP Tree (Reduced Error Pruning Tree)*: Algoritmo baseado em árvore de decisão, com valores fixos nas folhas;
2. *M5 Rules*: Algoritmo baseado em árvore de decisão com funções de regressão nas folhas, modificado para gerar um modelo de predição sob a forma de uma lista

de regras;

3. *Additive Regression*: Algoritmo de Meta-Aprendizado que realiza incremento ou decremento no valor de saída de acordo com o resultado obtido pelo algoritmo-base. Nos testes realizados, foi utilizado como algoritmo-base o *Decision Stump*;
4. *Isotonic Regression*: Algoritmo que toma como base o atributo que proporciona o menor erro quadrático, gerando a previsão de acordo com o intervalo de valores no qual se encontra esse atributo.

Como *baseline*, foram implementadas duas soluções. A primeira, chamada *Fixo*, operava em um canal único, o qual não variou nos experimentos. O canal de testes foi escolhido com base na frequência de operação das redes WLAN situadas na região, sendo escolhido o canal com maior quantidade de pontos de acesso. Já o segundo *baseline*, chamado *RSSI*, é baseado somente em energia, e foi inspirado no algoritmo proposto por Stabellini & Javed [2010]. Ele coleta 100 amostras de RSSI em cada canal sensoreado, somando-se os valores obtidos. O canal escolhido é aquele que apresenta a menor soma. A seguir é apresentado um passo-a-passo do funcionamento da solução *RSSI*.

1. Para cada canal:
 - a) O *Nó Central* coleta 100 amostras de RSSI;
 - b) É realizada a soma das amostras coletadas, obtendo-se a previsão para o respectivo canal;
2. O *Nó Central* seleciona o canal com melhor previsão, ou seja, com o menor soma de amostras de RSSI obtida;
3. O *Nó Central* transmite N pacotes de dados ao *Nó Auxiliar* localizado no canal selecionado, encaminhando o sumário do desempenho obtido ao computador, para mensurar o desempenho do algoritmo de troca de canal;
4. Reinicia-se o processo de decisão, retornando ao Passo 1.

O funcionamento desta solução é semelhante ao das soluções de Decisão, com diferença apenas no critério de escolha do melhor canal e no tempo de sensoreamento. Aqui, o tempo gasto pelas soluções de Decisão em cada canal também é reservado na solução baseada em RSSI, entretanto, caso o tempo gasto para coleta das amostras de RSSI seja inferior ao reservado por canal, não é realizada nenhuma operação até que esse tempo se esgote.

6.2 Primeiro Teste

O primeiro teste foi considerado preliminar, e foi baseado em somente dois canais de operação. Por isso, foram empregados dois nós sensores, sendo um *Nó Central* e um *Nó Auxiliar*. O *Nó Central* foi posicionado a uma distância de aproximadamente 30 centímetros do *Nó Auxiliar*. Por ser de configuração simplificada, exclusivamente neste cenário foi utilizado apenas um *Nó Auxiliar*, o qual acompanhava o *Nó Central* entre os dois canais sensoreados, de forma a realizar a transmissão e recepção de pacotes de sondagem.

Para a construção da base de treinamento, foi utilizado valor de α igual a 0,7 e tamanho da janela de previsão de aproximadamente seis minutos, assim como na base aplicada nos testes de sensibilidade (Seção 5.4). Com isso, foi definido um tempo de envio e recepção de pacotes de sondagem de aproximadamente um segundo (parâmetro T dos passos 1a e 1b da descrição do algoritmo de Decisão de Espectro apresentada na Seção 4.4). A única diferença entre a base de treinamento aqui construída e a que foi construída nos testes de sensibilidade está no intervalo de medição utilizado, o qual neste caso foi de cinco minutos para cada nó sensor.

Neste cenário, foi implementada apenas uma solução de Decisão, para fins de comparação com os *baselines*. Para a base de treinamento escolhida, o algoritmo *REP-Tree* (*Reduced Error Pruning Tree*), que é baseado em árvore de decisão, foi o que apresentou o melhor desempenho entre os 16 algoritmos analisados, segundo desempenho apresentado na ferramenta Weka, e portanto foi o algoritmo de AM escolhido para implementação do respectivo modelo de previsão. Cada uma das soluções implementadas foram executadas em uma repetição de uma hora. A cada intervalo de sensoreamento, eram enviados 10 pacotes de dados (valor de N do Passo 3 na descrição do algoritmo de Decisão de Espectro apresentada na Seção 4.4, e Passo 3 na descrição dos *baselines* apresentada na Seção 6.1) para mensurar o desempenho a cada escolha realizada.

Como fonte de interferência principal, foram colocados dois roteadores transmitindo dados entre si por meio do programa *iperf* [Russell & Cohn, 2012], gerando interferência em canais próximos aos da RSSF. Os roteadores se comunicavam durante 15 segundos, com intervalo de 10 segundos para troca de canal.

A Tabela 6.1 apresenta o desempenho obtido com as soluções implementadas, de acordo com taxa de entrega, o total de tentativas de transmissão, o total de colisões (erro do tipo ENACK), e a latência a cada pacote enviado.

Apesar do bom desempenho apresentado pela solução com Decisão de Espectro, em relação aos *baselines* e segundo todas as métricas analisadas, durante os experi-

Tabela 6.1. Resultados de desempenho: primeiro teste.

Solução Implementada	Taxa de Entrega (%)	Tentativas/Pacote	Colisões/Pacote	Latência/Pacote (ms)
REPTree	97,05	1,59	0,08	6,24
RSSI	95,53	1,86	0,13	6,83
Fixo	92,21	2,34	0,24	7,99

mentos foi observado que os valores de previsão gerados pelo algoritmo *REPTree* se resumiam em um pequeno conjunto de valores de saída, bem menor do que o número de valores presente na árvore gerada, ao contrário do que era esperado, pelo fato do cenário analisado apresentar alta interferência proveniente dos roteadores, com variabilidade elevada.

6.3 Segundo Teste

Aqui, procurou-se definir um cenário mais realista, com características diferentes dos cenários de coleta descritos na Seção 4.5.3. Para tanto, ao invés de dois canais de operação, foram considerados cinco canais, os quais não apresentavam sobreposição entre si, e portanto demandaram o uso de cinco *Nós Auxiliares*. Os *Nós Auxiliares* foram colocados distantes um do outro, a uma distância de aproximadamente cinco metros do *Nó Central*. Além disso, não foram utilizados roteadores para geração de interferência, como feito no primeiro teste descrito na Seção 6.2.

Com isso, temos um contexto diferente de tudo que havia sido feito até então, com o objetivo de testar o nível de generalização do modelo de previsão implementado. Como não era possível determinar qual seria o comportamento do meio ao longo do tempo neste teste, nem a qual magnitude, foi considerado importante que a granularidade sobre os valores de saída fosse elevada. Portanto, com base na baixa variabilidade entre os valores de saída observada no primeiro teste (Seção 6.2), foi implementado o segundo melhor algoritmo, chamado *M5Rules*.

O *M5Rules*, como descrito anteriormente, é um algoritmo de árvore modificado para que a mesma seja disposta na forma de uma lista de regras, cada uma associada a uma função de regressão específica. A definição de uma função de regressão para cada regra permite uma maior granularização do valor de saída, ao contrário do algoritmo *REPTree* que oferece apenas um conjunto específico de valores nas folhas da árvore de decisão.

A metodologia dos testes também foi diferente da que foi adotada no teste anterior. A solução baseada em Decisão (com o modelo de previsão obtido com o algoritmo

M5Rules) e os *baselines* implementados foram executadas em sequência, em quatro repetições de 15 minutos cada. A cada intervalo de sensoreamento, eram enviados 30 pacotes de dados, ao contrário de 10 pacotes como realizado no primeiro teste.

A Tabela 6.2 mostra os resultados de desempenho obtidos com cada uma das soluções implementadas, de acordo com taxa de entrega, o total de tentativas de transmissão, o total de colisões e a latência por pacote enviado. É apresentado o valor médio entre as quatro repetições realizadas, com intervalo de confiança de 95%.

Tabela 6.2. Resultados de desempenho: segundo teste.

Solução Implementada	Taxa de Entrega (%)	Tentativas/Pacote	Colisões/Pacote	Latência/Pacote (ms)
M5Rules	98,43	1,56	0,39	7,09
	(+/- 1,51)	(+/- 0,18)	(+/- 0,17)	(+/- 0,74)
RSSI	98,71	1,41	0,30	6,56
	(+/- 1,47)	(+/- 0,18)	(+/- 0,16)	(+/- 0,78)
Fixo	98,50	1,66	0,47	7,54
	(+/- 1,05)	(+/- 0,16)	(+/- 0,12)	(+/- 0,66)

Por meio da Tabela 6.2, observa-se que a solução com Decisão de Espectro, aqui representada pelo algoritmo *M5Rules*, não apresentou desempenho satisfatório, quando comparado com a solução baseada em RSSI e a solução baseada em canal Fixo. Isso mostra que o modelo de predição gerado pelo algoritmo *M5Rules* não proporcionou uma generalização suficiente para interpretar o comportamento dos canais, em um cenário diferente daqueles onde os dados das bases de treinamento foram coletados.

Com isso, fica evidente a necessidade de uma construção das bases de treinamento orientada à aplicação desejada, com a atualização periódica dos modelos de predição. O uso de um intervalo de medição reduzido, igual ou inferior a cinco minutos, como realizado neste teste, possibilita a criação de um sistema que exerça tal função a baixo custo temporal e computacional, sem interferências na aplicação principal.

6.4 Terceiro Teste

Em razão das deficiências de generalização observadas no segundo teste (Seção 6.3), com o objetivo de testar mais algoritmos de AM, foram coletados dados de desempenho sob condições semelhantes, com os nós distantes um do outro, caracterizando assim um *Quarto Cenário de Coleta*. Por meio dos dados coletados, juntamente com aqueles já coletados no Capítulo 4, foi criada uma nova base de treinamento, a qual também foi utilizada nos testes de sensibilidade da Seção 5.4. A única diferença em relação à base

aplicada nos dois testes anteriores (seções 6.2 e 6.3), além do novo cenário de coleta de dados foi que, ao invés de cinco minutos foram utilizados apenas dois minutos de meio de medições de cada nó, de forma que o número de instâncias fosse aproximado.

Executando novamente os 16 algoritmos de AM considerados na ferramenta Weka, utilizando para tal a nova base de treinamento, foi obtido o mesmo *ranking* de algoritmos da base anterior, dispensando a necessidade de um re-ajuste dos parâmetros α , tamanho da janela de previsão e tamanho da base de treinamento, realizado na Seção 5.2.

Neste teste avaliamos o comportamento da RSSF em um ambiente com baixa interferência de outras redes, semelhante ao que foi realizado no segundo teste (Seção 6.3). Aqui, os *Nós Auxiliares* foram posicionados a aproximadamente seis metros do *Nó Central*, sendo utilizada como fonte de interferência principal as redes WLAN existentes nas proximidades do ambiente de testes. A Figura 6.1 mostra as frequências de operação das redes WLAN e das redes *ZigBee*, a qual foi utilizada como base para determinar os canais próximos entre ambas. A Figura 6.2, por sua vez, mostra a intensidade de sinal das redes WLAN locais durante a realização dos testes.

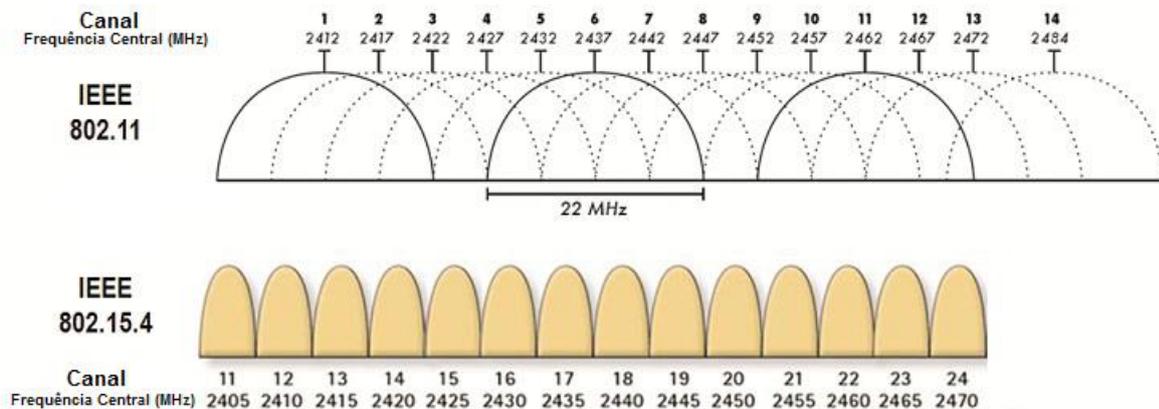


Figura 6.1. Frequências de operação das redes WLAN e das redes *ZigBee* [National Instruments, 2013].

As soluções de Decisão e os *baselines* foram executados em sequência durante 15 minutos cada, sendo realizadas cinco repetições. A Tabela 6.3 apresenta o desempenho obtido com as soluções implementadas, de acordo com taxa de entrega, o total de tentativas de transmissão, o total de colisões e a latência média a cada pacote enviado. Os resultados obtidos são apresentados com intervalo de confiança de 95%.

Com base nos resultados da Tabela 6.3, apenas uma das soluções de Decisão (*Additive Regression*) se mostrou melhor em relação à solução baseada em canal fixo, para todas as métricas analisadas. Para as outras soluções de Decisão, com exceção

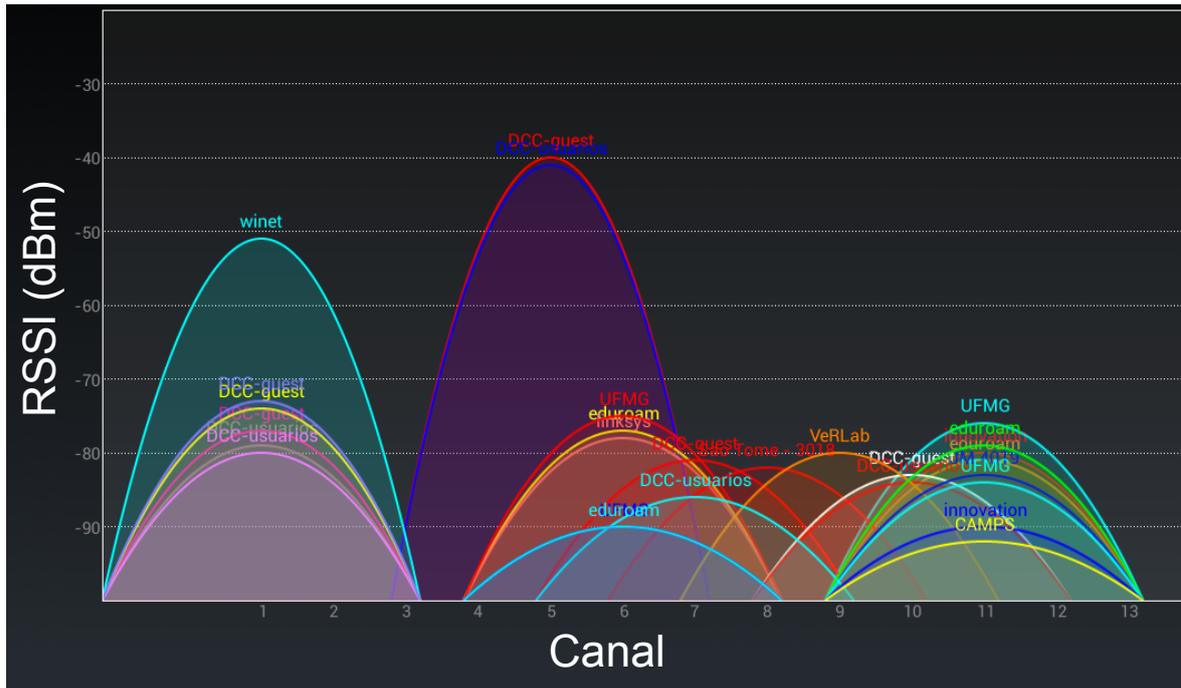


Figura 6.2. Redes WLAN em operação durante a realização dos testes.

Tabela 6.3. Resultados de desempenho: terceiro teste.

Solução Implementada	Taxa de Entrega (%)	Tentativas/Pacote	Colisões/Pacote	Latência/Pacote (ms)
REP Tree	96,44 (+/- 5,29)	1,50 (+/- 0,32)	0,37 (+/- 0,41)	6,78 (+/- 1,64)
M5 Rules	97,14 (+/- 2,33)	1,49 (+/- 0,18)	0,27 (+/- 0,18)	6,62 (+/- 0,73)
Additive Regression	98,88 (+/- 1,55)	1,29 (+/- 0,18)	0,14 (+/- 0,07)	5,80 (+/- 0,42)
Isotonic Regression	93,01 (+/- 16,48)	1,54 (+/- 0,82)	0,46 (+/- 0,87)	7,13 (+/- 3,70)
RSSI	97,50 (+/- 2,09)	1,43 (+/- 0,25)	0,25 (+/- 0,15)	6,37 (+/- 0,85)
Fixo	92,13 (+/- 6,15)	2,27 (+/- 0,75)	0,55 (+/- 0,34)	8,78 (+/- 2,04)

do *Isotonic Regression*, houve melhora segundo o total de tentativas de transmissão por pacote, ainda em relação à solução baseada em canal fixo. Além do total de tentativas de transmissão, o algoritmo *M5Rules* também foi superior ao mesmo *baseline* segundo a latência média por pacote. Nestes casos, foi possível verificar que as soluções

propostas foram capazes de detectar variações de comportamento das redes WLAN locais, acarretando na escolha de canais distintos e melhorando o desempenho, ao contrário da solução baseada em canal fixo que esteve sujeita às variações do canal utilizado.

Ao contrário do que foi observado no segundo teste (Seção 6.3), a superioridade da solução de Decisão *M5Rules*, em relação à solução baseada em canal fixo, sobre duas das quatro métricas analisadas, demonstrou a contribuição do quarto cenário de coleta incluído na nova base de treinamento, o qual foi descrito no início desta Seção. Observou-se também uma equivalência estatística entre as soluções de Decisão por meio do teste t de Student, o qual usa o intervalo de confiança apresentado entre parênteses na Tabela 6.3.

Por outro lado, a desvantagem de todas as soluções de Decisão em relação à solução baseada em RSSI se deve ao fato de que a fonte de interferência principal, no cenário analisado, se estabeleceu nas redes WLAN locais, as quais apresentaram poucas mudanças na sua configuração, como a potência de transmissão e o canal de operação, por exemplo. Além disso, essas redes atuaram em baixa magnitude sobre a RSSF, durante os experimentos realizados. As soluções de Decisão têm de forma intrínseca, em seus modelos de predição, interferências externas de maior magnitude, que acarretam em impactos consideráveis no desempenho da transmissão. Por isso, a aplicação dessas soluções em cenários onde a fonte de interferência é de menor magnitude pode acarretar em erros na previsão em determinados momentos. Nesses casos, a simples leitura do RSSI permite uma conclusão adequada sobre a situação de cada canal, acarretando em boas escolhas.

6.5 Quarto Teste

Neste cenário avaliamos o comportamento das soluções de Decisão em um cenário de coexistência, onde a interferência de redes coexistentes é mais intensa. Os *Nós Auxiliares* foram colocados próximos do *Nó Central*, a uma distância de aproximadamente trinta centímetros. Como fonte de interferência principal, além das redes WLAN localizadas na região, foram colocados dois roteadores próximos da RSSF transmitindo dados entre si por meio do programa *iperf*, em alta potência e à taxa máxima. Os roteadores se comunicavam durante 15 segundos, e em seguida trocavam de canal de forma aleatória, escolhendo canais próximos a um dos cinco empregados na RSSF. Antes de voltar a transmitir, os roteadores aguardavam 10 segundos para que a troca de canal fosse realizada com sucesso.

A Tabela 6.4 apresenta o desempenho médio obtido com as soluções implementadas, em cinco repetições e intervalo de confiança de 95%. As métricas avaliadas foram a taxa de entrega, o total de tentativas de transmissão, o total de colisões e a latência por pacote enviado.

Tabela 6.4. Resultados de desempenho: quarto teste.

Solução Implementada	Taxa de Entrega (%)	Tentativas/Pacote	Colisões/Pacote	Latência/Pacote (ms)
REP Tree	93,94 (+/- 4,07)	1,96 (+/- 0,42)	0,26 (+/- 0,19)	7,33 (+/- 1,30)
M5 Rules	92,88 (+/- 2,05)	2,29 (+/- 0,52)	0,26 (+/- 0,08)	7,92 (+/- 0,99)
Additive Regression	92,36 (+/- 1,88)	2,23 (+/- 0,31)	0,30 (+/- 0,09)	7,93 (+/- 0,60)
Isotonic Regression	94,78 (+/- 3,54)	1,88 (+/- 0,59)	0,18 (+/- 0,11)	7,01 (+/- 1,33)
RSSI	89,09 (+/- 3,12)	2,86 (+/- 0,39)	0,35 (+/- 0,04)	9,23 (+/- 0,68)
Fixo	88,55 (+/- 6,91)	2,90 (+/- 0,76)	0,36 (+/- 0,11)	9,34 (+/- 1,60)

Ao contrário do que foi observado no terceiro teste (Seção 6.4), todas as soluções de Decisão implementadas se mostraram superiores à solução baseada em RSSI, segundo todas as métricas analisadas. Com isso, verificamos a importância de se considerar uma gama maior de atributos durante o sensoreamento, como realizado nas soluções de Decisão, quando variações de comportamento do meio de maior magnitude são observadas. Nesses casos, a simples leitura do RSSI acarreta em escolhas ruins em determinados momentos, uma vez que a relação sinal-ruído varia consideravelmente.

Outro fator apresentado por meio da Tabela 6.4 é a equivalência de desempenho entre as soluções de Decisão, por meio do intervalo de confiança, da mesma forma como foi observado nos resultados do terceiro teste. Apesar das diferenças de desempenho obtidas na ferramenta Weka, as imprecisões presentes nas medidas e a variação do meio podem explicar o desempenho similar na implementação em uma RSSF real. Isso também mostra que o fator predominante no desempenho prático de um algoritmo de Decisão de Espectro utilizando AM é a modelagem dos dados da base de treinamento, e não a escolha do melhor algoritmo de AM segundo o desempenho obtido na ferramenta de análise, ou o ajuste fino dos seus parâmetros. Portanto, vemos que para o caso de

Decisão de Espectro em RSSF é mais interessante selecionar o algoritmo de AM que apresenta menor consumo de recursos computacionais.

6.6 Consumo de Memória das Soluções Implementadas

A Tabela 6.5 mostra o consumo de memória ROM e RAM, em *bytes*, para cada uma das soluções finais implementadas no terceiro e no quarto teste. Todos os algoritmos foram implementados utilizando ponto flutuante, o qual não possui uma unidade implementada em *hardware* na plataforma Iris, sendo portanto emulado em *software*. A plataforma oferece precisão de seis casas decimais, a qual também foi utilizada na base de treinamento escolhida para criação dos modelos de predição a serem implementados.

Tabela 6.5. Consumo de memória para cada solução implementada.

Solução	Consumo de Memória (<i>bytes</i>)	
	ROM	RAM
M5 Rules	46.492	1.284
REP Tree	24.638	1.284
Additive Regression	24.308	1.284
Isotonic Regression	24.306	1.284
RSSI	22.812	1.363
Fixo	21.950	1.271
Capacidade Total	128 K	8 K

Como esperado, a solução *Fixo* foi a que apresentou menor consumo de memória, uma vez que este não possui nenhum modelo de predição implementado. Por outro lado, a solução *M5Rules* foi a que apresentou o maior consumo, uma vez que este é baseado em árvore com funções de regressão nas folhas, como dito anteriormente, e por isso precisa armazenar uma quantidade extensa de números em ponto flutuante. As outras soluções apresentaram consumo de memória semelhante, principalmente pelo fato de trabalharem com um montante de valores aproximado entre eles.

6.7 Conclusão

Neste capítulo foram avaliadas as implementações realizadas em nós sensores reais. Foram escolhidos, dentre os melhores algoritmos testados na ferramenta Weka, aqueles

que eram passíveis de implementação, ou seja, que se encaixavam adequadamente nos recursos limitados de memória e processamento da plataforma utilizada. Foram realizados testes sob diferentes cenários, com o objetivo principal de verificar a capacidade real dos modelos de predição criados.

Verificou-se que as soluções baseadas em Decisão aumentam de fato o desempenho da RSSF, em relação a métodos baseados em canal fixo e baseados em energia, segundo todas as métricas consideradas. Entretanto, essa melhora se deu principalmente em cenários de coexistência, onde a variação no comportamento do meio é de maior magnitude, como ocorrido no primeiro e no quarto teste.

Capítulo 7

Conclusões

Tendo em vista a poluição da faixa ISM e as restrições de energia em RSSFs, este trabalho propôs o uso de algoritmos de AM de baixo custo computacional para a seleção do melhor canal de transmissão. Foram construídos modelos de regressão, com base em dados coletados a partir de plataformas reais. Estes modelos estimam o desempenho das transmissões a curto e longo prazo, permitindo a escolha do canal com melhor qualidade de comunicação para uso na RSSF.

Por meio da abordagem de regressão definiu-se, como valor de saída para os algoritmos de AM, a soma do número de tentativas de transmissão dos pacotes subsequentes, atributo que permitiu aproximar de forma simplificada o tempo necessário para a transmissão bem sucedida de uma sequência de pacotes em um determinado canal. Por meio de média móvel aplicada aos atributos de entrada, incorporou-se o comportamento histórico do canal em um único conjunto de atributos, simplificando as bases de treinamento a serem aplicadas nos algoritmos de AM. Com isso, foi possível criar modelos de predição mais simples e passíveis de implementação em uma plataforma de RSSFs comercial.

Um dos principais fatores observados, por meio dos testes de sensibilidade, foi a melhora de desempenho dos algoritmos de AM em geral, ainda que pequena, com a remoção de cada um dos atributos de entrada. Isso mostra a importância de se realizar uma avaliação prévia para descobrir o melhor conjunto de atributos de entrada, para cada um dos algoritmos de AM escolhidos para implementação em plataforma real, de forma simplificar ainda mais os modelos de predição gerados.

Verificou-se, por meio dos testes práticos, que o uso de AM aumenta o desempenho da RSSF em relação a métodos de canal fixo e de decisão baseada em energia, principalmente em cenários de coexistência, onde a variação no comportamento do meio é de maior magnitude, ou seja, quando o desempenho médio das transmissões

sofre impacto considerável. Isso reitera a aplicação final destinada a esses algoritmos, onde o meio se modifica constantemente, devido à existência de múltiplas redes no espectro.

7.1 Trabalhos Futuros

Como trabalhos futuros, pretende-se refinar o ajuste dos parâmetros dos modelos, adicionando também mecanismos para que o número de canais sensoreados seja reduzido a longo prazo. A redução no número de canais é importante porque permite uma simplificação ainda maior na complexidade da solução de Decisão aqui proposta, reduzindo o tempo de sensoreamento e o consumo de memória. Um dos critérios de eliminação de canais seria, por exemplo, um limiar que definisse um canal como congestionado por um tempo prolongado, o que dispensaria sua análise durante um intervalo também pré-definido, ou até mesmo ajustável de acordo com a solução de Decisão implementada.

Outro fator que permitiria uma redução na frequência de troca de canais, e que deve ser estudado posteriormente, é a definição de fatores de tolerância que justifiquem a troca de canal. Em outras palavras, uma vez que for detectado um canal cuja previsão é melhor do que aquela realizada sobre o canal de operação atual, é necessário definir um limiar que informe ao algoritmo de Decisão de Espectro o quão melhor outro canal deve ser, para que a troca seja realizada.

Pretende-se também investigar algoritmos de AM semi-supervisionados ou outras formas para a atualização periódica do modelo de predição. O uso de algoritmos Semi-Supervisionados oferece maior flexibilidade para esse tipo de atualização, o que complementaria as soluções aqui propostas, melhorando o seu desempenho.

Uma vez que neste trabalho consideramos apenas cenários onde os nós comunicam entre si em um salto, não foram utilizados protocolos de roteamento. Por isso, pretende-se também avaliar o impacto das soluções propostas sobre os principais protocolos de roteamento utilizados atualmente em plataformas comerciais de RSSFs, verificando assim a compatibilidade dessas soluções com as outras camadas da pilha de protocolos.

Referências Bibliográficas

- Akyildiz, I.; Lee, W.-Y. & Chowdhury, K. (2009). Spectrum management in cognitive radio ad hoc networks. *Network, IEEE*, 23(4):6–12. ISSN 0890-8044.
- Akyildiz, I. F.; Lo, B. F. & Balakrishnan, R. (2011). Cooperative spectrum sensing in cognitive radio networks: A survey. *Phys. Commun.*, 4(1):40–62. ISSN 1874-4907.
- Ansari, J.; Ang, T. & Mähönen, P. (2011). Wispot: Fast and reliable detection of wi-fi networks using ieee 802.15.4 radios. Em *Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '11, pp. 35–44, New York, NY, USA. ACM.
- Ansari, J. & Mähönen, P. (2010). Channel selection in spectrum agile and cognitive mac protocols for wireless sensor networks. Em *Proceedings of the 8th ACM international workshop on Mobility management and wireless access*, MobiWac '10, pp. 83–90, New York, NY, USA. ACM.
- Atmel (2006). *ZigBee™/IEEE 802.15.4-Transceiver - AT86RF230 Datasheet*. Atmel Corporation. 50p.
- Baldo, N.; Reddy, T. B.; Manoj, B. S.; Rao, R. & Zorzi, M. (2009). A neural network based cognitive controller for dynamic channel selection. Em *ICC*, pp. 1–5. IEEE.
- Balieiro, A. M.; Dias, K. L.; Castro, A. R. G. & Alves, E. R. (2010). Handoff de espectro em redes baseadas em rádio cognitivo utilizando redes neurais artificiais. Congresso da Sociedade Brasileira de Computação (CSBC 2010).
- Bantouna, A.; Stavroulaki, V.; Kritikou, Y.; Tsagkaris, K.; Demestichas, P. & Moessner, K. (2012). An overview of learning mechanisms for cognitive systems. *EURASIP J. Wireless Comm. and Networking*, 2012:22.
- Clancy, C.; Hecker, J.; Stuntebeck, E. & O'Shea, T. (2007). Applications of machine learning to cognitive radio networks. *IEEE Wireless Communications*, 14(4):47–52.

- Correia, L. H. A.; Oliveira, E. E.; Macedo, D. F.; Moura, P.; Loureiro, A. A. F. & Silva, J. S. (2012). A framework for cognitive radio wireless sensor networks. Em *ISCC*, pp. 611–616. IEEE.
- Crossbow (2003). *Mica2 Datasheet*. Crossbow Technology. 2p.
- Crossbow (2008a). *Iris Datasheet*. Crossbow Technology. 2p.
- Crossbow (2008b). *MicaZ Datasheet*. Crossbow Technology. 2p.
- Culler, D. E.; Estrin, D. & Srivastava, M. B. (2004). Guest editors' introduction: Overview of sensor networks. *IEEE Computer*, 37(8):41–49.
- Denkovski, D.; Atanasovski, V. & Graviļovska, L. (2011). Efficient mid-end spectrum sensing implementation for cognitive radio applications based on usrp2 devices. Em *The First International Conference on Advances in Cognitive Radio (COCORA 2011)*, Budapest, Hungary.
- FCC (2002). Spectrum policy task force report. Federal Communications Commission, Federal Communications Commission (FCC'02), Washington, DC, USA, November.
- FCC (2005). Notice of proposed rule making and order: Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies. ET Docket No. 03-108.
- Felice, M. D.; Chowdhury, K. R. & Bononi, L. (2011). Learning with the bandit: A cooperative spectrum selection scheme for cognitive radio networks. Em *GLOBECOM*, pp. 1–6. IEEE.
- Felice, M. D.; Chowdhury, K. R.; Wu, C.; Bononi, L. & Meleis, W. (2010). Learning-based spectrum selection in cognitive radio ad hoc networks. Em Osipov, E.; Kassler, A.; Bohnert, T. M. & Masip-Bruin, X., editores, *WWIC*, volume 6074 of *Lecture Notes in Computer Science*, pp. 133–145. Springer.
- Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P. & Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18. ISSN 1931-0145.
- Jayaweera, S.; Li, Y.; Bkassiny, M.; Christodoulou, C. & Avery, K. (2011). Radiobots: The autonomous, self-learning future cognitive radios. Em *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pp. 1–5.

- Katidiotis, A.; Tsagkaris, K. & Demestichas, P. (2010). Performance evaluation of artificial neural network-based learning schemes for cognitive radio systems. *Computers & Electrical Engineering*, 36(3):518–535.
- Khaleel, H.; Penna, F.; Pastrone, C.; Tomasi, R. & Spirito, M. (2010). Distributed spectrum sensing and channel selection in opportunistic wireless personal area networks. Em *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, MobiOpp '10, pp. 185–187, New York, NY, USA. ACM.
- Khaleel, H.; Penna, F.; Pastrone, C.; Tomasi, R. & Spirito, M. (2012). Frequency agile wireless sensor networks: Design and implementation. *Sensors Journal, IEEE*, 12(5):1599–1608. ISSN 1530-437X.
- Kim, K.; Xin, Y. & Rangarajan, S. (2010). Energy detection based spectrum sensing for cognitive radio: An experimental study. Em *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5. ISSN 1930-529X.
- Lee, W. Y.; Chowdhury, K. R. & Vuran, M. (2008). *Cognitive Radio Networks*, capítulo Spectrum Sensing Algorithms for Cognitive Radio Networks. CRC Press. Edited by Y. Xiao and F. Hu.
- Liu, T. & Cerpa, A. (2011). Foresee (4c): Wireless link prediction using link features. Em Koutsoukos, X. D.; Langendoen, K.; Pottie, G. J. & Raghunathan, V., editores, *IPSN*, pp. 294–305. IEEE.
- Liu, Y.; Tamma, B.; Manoj, B. & Rao, R. (2010). Traffic prediction for cognitive networking in multi-channel wireless networks. Em *INFOCOM IEEE Conference on Computer Communications Workshops*, 2010, pp. 1–6.
- Marinho, J. & Monteiro, E. (2012). Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions. *Wirel. Netw.*, 18(2):147–164. ISSN 1022-0038.
- Masonta, M.; Mzyece, M. & Ntlatlapa, N. (2013). Spectrum decision in cognitive radio networks: A survey. *Communications Surveys Tutorials, IEEE*, 15(3):1088–1107. ISSN 1553-877X.
- Meshkova, E.; Ansari, J.; Denkovski, D.; Riihijärvi, J.; Nasreddine, J.; Pavloski, M.; Gavrilovska, L. & Mähönen, P. (2011). Experimental spectrum sensor testbed for constructing indoor radio environmental maps (poster paper). Em *IEEE DySPAN*, pp. 603–607.

- MoteIv (2006). *Tmote Sky Datasheet*. MoteIv Corporation. 28p.
- National Instruments (2013). National instruments. www.ni.com. Acesso em: Agosto/2013.
- Open Automation (2013). Open automation. www.openautomation.net. Acesso em: Janeiro/2013.
- Pal, A. & Nasipuri, A. (2012). A distributed channel selection scheme for multi-channel wireless sensor networks. Em Nasipuri, A.; Hu, Y.-C. & Shakkottai, S., editores, *MobiHoc*, pp. 263–264. ACM.
- Russell, J. & Cohn, R. (2012). *Iperf*. Book on Demand. ISBN 9785512527092.
- Shi, X. & de Francisco, R. (2011). Adaptive spectrum sensing for cognitive radios: An experimental approach. Em *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pp. 1408 –1413. ISSN 1525-3511.
- Song, C.; Rahman, M. & Harada, H. (2011). Proposal and hardware implementation of smart threshold setting methods for spectrum sensing. Em *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pp. 918 –923.
- Song, M.; Xin, C.; Zhao, Y. & Cheng, X. (2012). Dynamic spectrum access: from cognitive radio to network radio. *Wireless Communications, IEEE*, 19(1):23 –29. ISSN 1536-1284.
- Stabellini, L. & Javed, M. (2010). Experimental comparison of dynamic spectrum access techniques for wireless sensor networks. Em *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pp. 1 –5. ISSN 1550-2252.
- Taj, M. & Akil, M. (2011). Cognitive radio spectrum evolution prediction using artificial neural networks based multivariate time series modelling. Em *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, pp. 1–6, 27–29.
- Tamma, B. R.; Manoj, B. S. & Rao, R. R. (2012). Traffic sensing and characterization in multi-channel wireless networks for cognitive networking. *Comput. Netw.*, 56(7):1968--1982. ISSN 1389-1286.
- Tsagkaris, K.; Bantouna, A. & Demestichas, P. (2012). Self-organizing maps for advanced learning in cognitive radio systems. *Comput. Electr. Eng.*, 38(4):862--881. ISSN 0045-7906.

- Tsagkaris, K.; Katidiotis, A. & Demestichas, P. (2008). Neural network-based learning schemes for cognitive radio systems. *Computer Communications*, 31(14):3394–3404.
- Unnikrishnan, J. & Veeravalli, V. V. (2010). Algorithms for dynamic spectrum access with learning for cognitive radio. *IEEE Transactions on Signal Processing*, 58(2):750–760.
- Wang, Q. & Zheng, H. (2006). Route and spectrum selection in dynamic spectrum networks. Em *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pp. 625 – 629.
- Yang, Y.; Liu, Y.; Zhang, Q. & Ni, L. (2010). Cooperative boundary detection for spectrum sensing using dedicated wireless sensor networks. Em *INFOCOM, 2010 Proceedings IEEE*, pp. 1 –9. ISSN 0743-166X.
- Yarkan, S.; Halbawi, W. & Qaraqe, K. A. (2011a). An experimental setup for performance evaluation of spectrum sensing via energy detector: indoor environment. Em Frattasi, S. & Marchetti, N., editores, *CogART*, p. 42. ACM.
- Yarkan, S.; Toreyin, B.; Qaraqe, K. & Cetin, A. (2011b). An experimental validation of an online adaptive cooperation scheme for spectrum sensing. Em *Telecommunications (ICT), 2011 18th International Conference on*, pp. 226 –231.
- Yu, R.; Zhang, Y.; Yi, L.; Xie, S.; Song, L. & Guizani, M. (2012). Secondary users cooperation in cognitive radio networks: balancing sensing accuracy and efficiency. *Wireless Communications, IEEE*, 19(2):30 –37. ISSN 1536-1284.
- Yucek, T. & Arslan, H. (2009). A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys Tutorials, IEEE*, 11(1):116 –130. ISSN 1553-877X.
- Zeng, Y.; Liang, Y.-C.; Hoang, A. T. & Zhang, R. (2010). A review on spectrum sensing for cognitive radio: challenges and solutions. *EURASIP J. Adv. Signal Process*, 2010:2:2--2:2. ISSN 1110-8657.
- Zhao, Q. & Swami, A. (2007). A survey of dynamic spectrum access: Signal processing and networking perspectives. Em *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pp. IV–1349 –IV–1352. ISSN 1520-6149.