

**APLICAÇÕES DE APRENDIZAGEM DE
MÁQUINA NA ESTIMAÇÃO DE QUALIDADE
DE ENLACE EM COMUNICAÇÕES SEM FIO**

WENDLEY SOUZA DA SILVA

APLICAÇÕES DE APRENDIZAGEM DE
MÁQUINA NA ESTIMAÇÃO DE QUALIDADE
DE ENLACE EM COMUNICAÇÕES SEM FIO

Tese apresentada ao Programa de Pós-
-Graduação em Ciência da Computação
do Instituto de Ciências Exatas da Uni-
versidade Federal de Minas Gerais como
requisito parcial para a obtenção do grau
de Doutor em Ciência da Computação.

ORIENTADOR: JOSÉ MARCOS S. NOGUEIRA
COORIENTADOR: DANIEL FERNANDES MACEDO,
MICHELE NOGUEIRA (UFPR)

Belo Horizonte - MG

Novembro de 2020

© 2020, Wendley Souza da Silva.
Todos os direitos reservados.

Silva, Wendley Souza da.

S586a Aplicações de aprendizagem de máquina na estimação de qualidade de enlace em comunicações sem fio [manuscrito] / Wendley Souza da Silva.- 2020.
xvi, 108 f. il.

Orientador: José Marcos Silva Nogueira.
Coorientador: Daniel Fernandes Macedo.
Coorientadora: Michele Nogueira Lima.
Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f.98-108.

1. Computação – Teses. 2. Estimação de qualidade de enlace – Teses. 3. Redes de sensores sem fio – Teses. 4. Aprendizado de máquina – Teses. I. Nogueira, José Marcos Silva. II. Macedo, Daniel Fernandes. III. Lima, Michele Nogueira. IV. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. V. Título.

CDU 519.6*22(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa CRB
6ª Região nº 1510



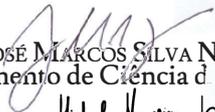
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE POS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Aplicações de aprendizagem de máquina na estimação de qualidade de enlace
para comunicações sem fio

WENDLEY SOUZA DA SILVA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

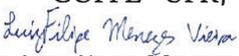

PROF. JOSÉ MÁRCOS SILVA NOGUEIRA - Orientador
Departamento de Ciência da Computação - UFMG


PROF^a. MICHELE NOGUEIRA LIMA - Coorientadora
Departamento de Informática - UFPR


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


PROF. FRANCISCO RODRIGO PORTO CAVALCANTI
Centro de Tecnologia - UFC


PROF. JOSÉ FERREIRA DE REZENDE
COPPE - UFRJ


PROF. LUIZ FILIPE MENEZES VIEIRA
Departamento de Ciência da Computação - UFMG


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

Belo Horizonte, 26 de Novembro de 2020.

Dedico este trabalho primeiramente ao meu avô Ivan Paulo (in memoriam), cuja presença foi importante em minha vida. Dedico aos meus filhos Adler e Elisa, à minha esposa Iracema, e aos meus pais, Fátima e Assis.

Agradecimentos

Agradeço inicialmente a Deus por fornecer a saúde, sabedoria e energia a mim e a todas as pessoas mencionadas adiante.

À minha família, especialmente meus pais Francisco de Assis e Fátima, que muito fizeram para que eu pudesse ter uma educação formal adequada em momentos difíceis; à minha querida esposa Iracema, pelo incansável apoio, carinho e compreensão ao longo dessa jornada; sem você ao meu lado dificilmente superaria as dificuldades. Aos meus filhos Adler e Elisa que, mesmo sem muito perceberem e a seus modos, compreendem minhas necessárias eventuais ausências e me ensinam sobre paciência e produtividade. À minha irmã, Karen, pelo contínuo incentivo à formação; à minha sogra, Suzana, pelo importante apoio durante anos do doutoramento.

Aos meus dedicados orientador e coorientadores, professores José Marcos, Daniel Macedo e Michele Nogueira, pelas incontáveis horas dedicadas em transmitir conhecimento com atenção, pelas precisões de suas observações e pela imensa paciência ao longo dessa jornada. Todos me ensinaram como ser um estudante e um ser humano melhor. Os profs. José Marcos, Daniel e Michele já eram para mim modelo de ótimos profissionais e, ao longo do doutoramento, se tornaram também referência de professores competentes, profissionais, atenciosos e humanos.

À professora Thi Mai Trang Nguyen (UPMC - Sorbonne Université, Paris), pela grande contribuição nas discussões que propiciaram a evolução do trabalho e pela importante colaboração em um trabalho científico; ao professor Marcos Augusto Vieira (*Flop*) pelas precisas e desafiantes observações quando discutíamos sobre os projetos no laboratório.

Aos meus colegas do laboratório Winet, com quem muitas vezes pude compartilhar e ouvir ideias incríveis sobre assuntos diversos.

Às funcionárias do DCC, principalmente Sônia e Linda, que sempre me atenderam de forma paciente e que em muitas ocasiões me explicaram mais de uma vez algum trâmite burocrático que eu estava em dúvida; aos professores do DCC,

principalmente aqueles com quem tive contato em disciplinas formais, pelo dedicado ensino e pelas lições e questionamentos sempre desafiadores, dentre os quais me recordo com muito carinho: professores Luiz Chaimowicz, Jussara Almeida e Newton Vieira.

À Universidade Federal do Ceará, principalmente aos professores do Curso de Engenharia da Computação (*campus* Sobral), pelo apoio e compreensão durante meu afastamento para o doutoramento.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), pelo imprescindível apoio financeiro durante o doutoramento; ao projeto EU-BR FUTEBOL e FAPEMIG, por terem propiciado a realização de muitos dos meus experimentos utilizando os rádios definidos por *software* mesmo eu estando fisicamente distante do laboratório.

*“Há tempo, muito tempo que eu estou longe de casa
E nessas ilhas cheias de distância o meu blusão de couro
Se estragou*

*(...) Gente de minha rua, como eu andei distante
Minha normalista linda, ainda sou estudante
Da vida que eu quero dar*

*E até parece que foi ontem minha mocidade
Com diploma de sofrer de outra Universidade
Minha fala nordestina, quero esquecer o francês*

*E vou viver as coisas novas que também são boas
O amor, humor das praças cheias de pessoas
Agora eu quero tudo, tudo outra vez.”*

(Antônio Carlos Belchior, compositor natural de Sobral-CE
Trecho da canção “Tudo outra vez”)

Resumo

A popularização dos dispositivos embarcados e a sua comunicação direta sem fio, *i.e.* comunicação máquina a máquina (M2M), é uma tendência crescente nas redes atuais. Um número cada vez maior de dispositivos embarcados suporta diferentes tipos de aplicações, tais como cuidados de saúde, vigilância, monitoramento de gás entre outros, que exigem um nível elevado de confiabilidade de comunicação, devido à informação crítica que carregam. No entanto, o uso de muitos desses dispositivos aumenta a concorrência para o espectro de frequência, dificultando a obtenção de uma comunicação confiável. Para superar essas limitações, o uso de estimadores de qualidade de enlace (LQE – do inglês *Link Quality Estimators*) é crucial para proporcionar uma comunicação eficiente. Com o objetivo de possibilitar uma comunicação robusta e rápida sob condições adversas, tais como ruídos, este trabalho propõe dois novos LQE, chamados PRR² e LQM, que utilizam múltiplas métricas de qualidade de enlace e diferentes estratégias para o cômputo da estimativa. O estimador PRR² capta variações de qualidade do enlace no curto prazo, mas também considera as variações de longo prazo. O estimador LQM utiliza técnicas de aprendizado de máquina com detecção e tratamento de *concept drift* para prover uma estimativa precisa. Ambos estimadores propostos são comparados com o estado-da-arte em um protótipo usando rádios USRPs (*Universal Software Radio Peripheral*), e os resultados mostram que as propostas aumentam a confiabilidade dos enlaces ao reduzir o número de retransmissões em até 61% e ao aumentar a taxa de entrega em até 43%.

Palavras-chave: Estimadores de qualidade de enlace, Comunicação máquina a máquina, aprendizagem de máquina, LQM, PRR².

Abstract

The popularization of embedded devices and their direct wireless communication, i.e., machine-to-machine (M2M) communication, is a growing trend in today's networks. An increasing number of embedded devices support different types of applications, such as healthcare, surveillance, gas monitoring, among others, which require a high level of communication reliability due to the critical information they carry. However, the use of many of these devices increases competition for the frequency spectrum, making it difficult to achieve reliable communication. To overcome these limitations, the use of link quality estimators (LQE - Link Quality Estimators) is crucial to provide efficient communication. In order to enable a robust and fast communication under adverse conditions, such as noise, this work proposes two new LQE, called PRR² and LQM, which use multiple link quality metrics and different strategies for calculating the estimate. The PRR² estimator captures variations in link quality in the short term, but also considers long-term variations. The LQM estimator uses machine learning techniques with concept drift detection and adaptation to provide an accurate estimate. Both proposed estimators are compared with the state-of-the-art in a prototype using USRPs (Universal Software Radio Peripheral) radios, and the results show that the proposals increase the reliability of the links by reducing the number of retransmissions by up to 61% and by increase the delivery rate by up to 43%.

Keywords: Machine-to-machine communication, link quality estimators, machine learning, PRR², LQM.

Lista de Figuras

1.1	Tráfego M2M mensal esperado para o ano 2022 no mundo, em comparação ao ano de 2017. Fonte dos dados: Cisco [2019]	3
1.2	Aplicabilidade da estimativa de qualidade do enlace	4
1.3	<i>Tradeoff</i> a ser considerado no processo de estimação de qualidade do enlace - adaptado de Lowrance & Lauf [2017]	7
2.1	Modelo de sistema de comunicação M2M - adaptado de Verma et al. [2016]	13
2.2	Etapas para estimativa de qualidade do enlace – adaptado de Baccour et al. [2012]	24
2.3	Diagrama de blocos de um sistema de controle - adaptado de Hellerstein et al. [2004]	25
2.4	Curva de resposta de um sistema de controle de processos - Hellerstein et al. [2004]	26
2.5	Classificação dos estimadores de qualidade de enlace	28
2.6	Programação tradicional – adaptado de [Lee, 2019]	33
2.7	Aprendizagem de máquina – adaptado de [Lee, 2019]	33
2.8	Modelo de processo supervisionado de aprendizagem de máquina – adaptado de [Jones, 2017]	34
2.9	Modelo de processo não supervisionado de aprendizagem de máquina – adaptado de [Jones, 2017]	34
2.10	Modelo de processo de aprendizagem por reforço – adaptado de [Jones, 2017]	35
2.11	Exemplo de classificação – adaptado de [Lee, 2019]	37
2.12	Exemplo de regressão – adaptado de [Lee, 2019]	37
3.1	Taxonomia dos estimadores discutidos nesta seção	41
4.1	Esquema de PRR ² : janelas PRR curta e longa	48
4.2	Entradas e saída do estimador proposto PRR ²	49

4.3	Etapas de operação do PRR ²	51
4.4	Diagrama de sequência do PRR ²	54
5.1	Esquema de operação do LQM	56
5.2	Fluxo de funcionamento do emissor com LQM	57
5.3	KDE com linha da forma gaussiana: KDE do RSSI na imagem superior e KDE do SNR na imagem inferior	61
5.4	Função de autocorrelação	63
5.5	Correlação de Pearson para 8 <i>features</i>	64
5.6	Correlação de Spearman para 8 <i>features</i>	65
6.1	Entrada e saída do módulo de controle de potência	74
6.2	Relação entre os valores estimados e o ganho de potência da transmissão	75
6.3	Disposição dos USRPs durante um experimento com o PRR ²	76
6.4	Comparações para taxa de entrega	78
6.5	Transmissões por pacotes recebidos	79
6.6	Gain index da transmissão	79
6.7	Taxa de entrega	80
6.8	Número de retransmissões por pacote recebido	81
6.9	Ganho na transmissão	82
6.10	Taxa de entrega no segundo cenário	83
6.11	Quantidade de retransmissões por pacote recebidos no segundo cenário .	84
6.12	Ganho na transmissão no segundo cenário	84
6.13	Disposição dos USRPs para o experimento de avaliação do LQM	86
6.14	Taxa de entrega e TPRP sem adição de ruído artificial	88
6.15	TGI e custo energético sem adição de ruído artificial	89
6.16	Taxa de entrega e TPRP com adição de ruído artificial	90
6.17	TGI e custo de energia com adição de ruído artificial	91
6.18	Taxa de entrega - cenário com múltiplos remetentes	92
6.19	Taxa de entrega e TPRP - cenário de múltiplos remetentes agregados . .	93
6.20	TGI e custo energético - cenário de múltiplos remetentes agregado	93

Lista de Tabelas

2.1	Requisitos de tráfego M2M conforme a aplicação - adaptado de Verma et al. [2016]	16
3.1	Tabela comparativa de LQEs	45
5.1	Teste estatístico ADF	62
5.2	Matriz de correlação de Pearson	66
5.3	Desempenho dos algoritmos de <i>Machine Learning</i> - <i>target</i> : taxa de Entrega	68
5.4	Desempenho dos algoritmos de <i>Machine Learning</i> - <i>target</i> : TPRP	69
5.5	Hiperparâmetros utilizados na ML	69
5.6	Tempo de treinamento (<i>cross validation</i> e <i>shuffle split</i>)	70
6.1	Médias e intervalos de confiança da taxa de entrega	77
6.2	Descrição dos cenários do experimento com LQM	87

Sumário

Agradecimentos	vi
Resumo	ix
Abstract	x
Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Motivação	2
1.2 Definições preliminares	5
1.3 Problemática	6
1.4 Objetivos	7
1.5 Contribuições	8
1.6 Organização do trabalho	9
2 Fundamentos teóricos	11
2.1 Comunicação Máquina a Máquina	11
2.1.1 Modelos de comunicação M2M	12
2.1.2 Tipos de comunicação	14
2.1.3 Tráfego M2M	15
2.1.4 Arquiteturas e padrões M2M	16
2.1.5 Requisitos e características das comunicações M2M	18
2.1.6 Termos relacionados: D2D, IoT, WSN e CPS	19
2.2 Controle de Potência de Transmissão	21
2.3 Estimadores de qualidade do enlace	22
2.3.1 Etapas	23

2.3.2	Requisitos	25
2.3.3	Classificação dos LQEs	27
2.3.4	Métricas baseadas em potência do sinal	27
2.3.5	Métricas baseadas em estatísticas de pacotes	29
2.4	Séries temporais	30
2.4.1	Técnicas de predição e suavização	30
2.5	Aprendizagem de máquina	32
2.5.1	Aprendizagem supervisionada	35
2.5.2	<i>Concept drift</i>	36
2.6	Conclusão	38
3	Trabalhos relacionados	40
3.1	LQEs baseados em única métrica ou camada de rede	40
3.2	LQEs baseados em informações de múltiplas camadas de rede	42
3.3	LQEs baseados em processos de aprendizagem de máquina	43
3.4	Conclusão	46
4	Estimador PRR²	47
4.1	Descrição do PRR ²	47
4.1.1	Parâmetros de entrada e saída	48
4.2	Funcionamento do PRR ²	50
4.3	Conclusão	53
5	Estimador LQM	55
5.1	LQM - estimador	55
5.2	Obtenção das entradas	58
5.3	Definição dos algoritmos de aprendizagem de máquina	59
5.3.1	Analisando a série temporal	60
5.3.2	Selecionando as entradas do modelo	62
5.3.3	Resultados do treinamento	66
5.3.4	Identificando <i>Concept Drift</i>	70
5.4	Conclusão	71
6	Resultados e Análise	72
6.1	Avaliação	72
6.2	Resultados dos experimentos com PRR ²	73
6.2.1	Implementação do PRR ² no USRP	73
6.2.2	Descrição do algoritmo de controle de potência	74

6.2.3	Cenários de experimentação	75
6.2.4	Resultados	76
6.2.5	Discussão	84
6.3	Resultados dos experimentos com LQM	85
6.3.1	Implementação do LQM no USRP	85
6.3.2	Resultados	87
6.3.3	Discussão	91
7	Considerações finais	94
7.1	Conclusões	94
7.2	Limitações	96
7.3	Trabalhos futuros	96
7.4	Agradecimentos	97
	Referências Bibliográficas	98

Capítulo 1

Introdução

Nos últimos anos, o uso em larga escala de dispositivos capazes de acessar redes sem fio, como *smartphones*, *tablets*, câmeras, *notebooks*, monitores e sensores de saúde e outras aplicações móveis causou um aumento na demanda por altas taxas de dados nas comunicações sem fio, apesar da escassa largura de banda [Axell et al., 2012]. As projeções da empresa Cisco sugerem que entre os anos de 2017 e 2022 ocorrerá um crescimento de 7 vezes na quantidade de tráfego de dados provenientes de dispositivos móveis inteligentes, na medida em que pessoas e indústrias aumentem o número de sensores e dispositivos inteligentes em uso [Cisco, 2019]. Isso implica maior concorrência pelo espectro de frequência de operação [Chen et al., 2017], acarretando uma possível redução da confiabilidade da comunicação sem fio e afetando a velocidade de transmissão quando muitos equipamentos estiverem em uso simultaneamente [Luo et al., 2015].

Conjuntamente, cada vez mais aplicativos críticos que requerem alta confiabilidade e desempenho fazem uso da comunicação sem fio [Alexiou & Gotsis, 2015], exigindo que a rede mantenha a comunicação para que eventuais limitações não afetem o funcionamento dos aplicativos. Por exemplo, a telemetria de carros em movimento, os equipamentos de segurança em uma rede doméstica e o monitoramento da pressão arterial e dos batimentos cardíacos em idosos, exigem alta disponibilidade, mesmo em áreas onde existe ruído eletromagnético significativo.

Um conceito comum a essa ampla conectividade doméstica, médica e industrial é geralmente nomeado como comunicação máquina a máquina (*Machine-to-Machine* - M2M). M2M envolve desde tecnologias capazes de medir e entregar dados até as comunicações entre processadores embutidos e sensores inteligentes, sem ou com o mínimo de interação humana durante os vários estágios de operação [Anton-Haro & Dohler, 2014]. Este trabalho focará no desenvolvimento de estimadores de qualidade

do enlace para comunicações M2M.

1.1 Motivação

De maneira crescente, as redes sem fio vêm sendo utilizadas para aplicações críticas que exigem alta confiabilidade e desempenho, o que, por vezes, requer funcionamento adequado mesmo em ambientes eletromagneticamente ruidosos, constituindo desafios significativos para os pesquisadores e engenheiros desenvolverem estratégias para superar essas dificuldades, como será discutido a seguir.

O funcionamento de vários dispositivos sem fio compartilhando o mesmo canal aumenta o ruído e a interferência no sinal de comunicação. Além disso, o uso crescente de aplicações M2M gera uma demanda por mais largura de banda, o que pode levar a redução da confiabilidade e do desempenho da rede [Zhang et al., 2012]. Tais fatores, conseqüentemente, requerem comunicações M2M de alto desempenho e que mantenham a confiabilidade mesmo em condições de interferência eletromagnética. Além disso, reduzir a quantidade de falhas ou incrementar as taxas de transmissão utilizando a comunicação sem fio é custoso porque a condição do canal de comunicação varia com o tempo [Verma et al., 2016]. Desta forma, um mecanismo que consiga ultrapassar esses obstáculos torna-se importante e útil.

Confiabilidade é a capacidade do sistema de realizar e manter a comunicação em condições preestabelecidas em um dado intervalo de tempo. Devido à natureza das transmissões em redes sem fio por difusão (*broadcast*), as comunicações são propensas a erros, o que acarreta em elevadas variações das taxas de bits errados (*Bit Error Rate* - BER) em um determinado intervalo de tempo [Nguyen et al., 2008]. Essas variações causam perdas ou atrasos no recebimento das mensagens, o que reduz a confiabilidade das redes. As comunicações não confiáveis podem resultar em mau funcionamento ou interrupção das aplicações M2M, por exemplo, em serviços de monitoramento de desastres ou de cuidados com a saúde. Desse modo, é fundamental compreender os fatores que impactam a confiabilidade de comunicação das redes M2M a fim de conceber técnicas e protocolos que minimizem ou evitem essas ocorrências.

Com o aumento da quantidade de dispositivos conectados à internet, há uma propensão para uma redução da confiabilidade da comunicação. Estudos especializados indicam que o tráfego M2M aumentará 8 vezes entre os anos de 2017 e 2022 [Cisco, 2019], representando uma taxa de crescimento anual composta (CAGR - *Compound Annual Growth Rate*) de 52%, e que em 2023 as conexões M2M re-

presentarão 50% de todos os dispositivos conectados, enquanto em 2018 eram 33% (CAGR de 19,2%) [Cisco, 2020]. Na Figura 1.1 está ilustrada essa expectativa de crescimento, indicando que no ano de 2022 o tráfego M2M global deve alcançar o volume de 77,5 Exabytes por mês. Para a América Latina, a mesma fonte projeta um crescimento similar, de 6 vezes, alcançando um volume mensal de 4,4 EB de tráfego M2M. Essa elevada taxa de expansão indica que serão necessários grandes esforços para oferecer soluções que suportem o número crescente de comunicações M2M com características e necessidades diferentes. Essas características criam problemas que demandam soluções específicas, conforme detalhados adiante na Seção 1.3.

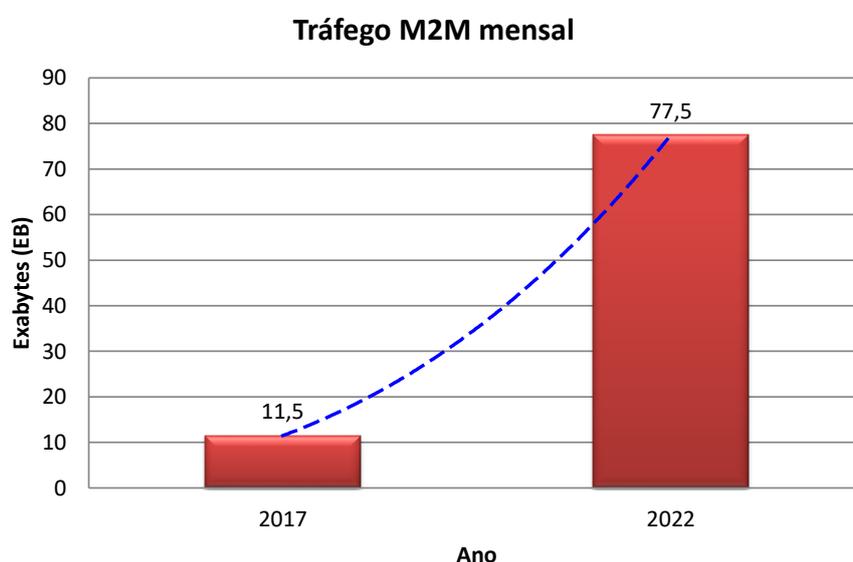


Figura 1.1. Tráfego M2M mensal esperado para o ano 2022 no mundo, em comparação ao ano de 2017. Fonte dos dados: Cisco [2019]

Estimadores de qualidade de enlace precisos são requisitos fundamentais para alcançar comunicações sem fio confiáveis e eficientes. Eficiência consiste em alcançar o resultado esperado com o mínimo de erros e de recursos empregados. Conforme relatado por De Couto et al. [2005], um método de estimação de qualidade do enlace impreciso ou lento pode causar uma redução superior a 50 % na vazão (*throughput*) da rede, principalmente devido ao algoritmo de roteamento operar de uma forma ineficiente. No entanto, devido à natureza dinâmica dos canais sem fio, métodos de

estimação energeticamente eficientes, precisos, reativos e estáveis continuam a ser uma tarefa desafiadora.

Muitos algoritmos tentam adaptar a comunicação às condições do canal, utilizando-se de mecanismos de controle de potência de transmissão [Xu et al., 2013], adaptação de taxa de bits [Biaz & Wu, 2008] ou até mesmo do estabelecimento de rotas usando a qualidade do enlace como principal parâmetro [Akkaya & Younis, 2005]. Para que todas essas estratégias tenham sucesso, é imprescindível a estimação precisa da qualidade de enlace (*Link Quality Estimation* - LQE). A estimação da qualidade do enlace consiste em aplicar técnicas de caracterização estatística de enlaces os quais empregam teorias de predição.

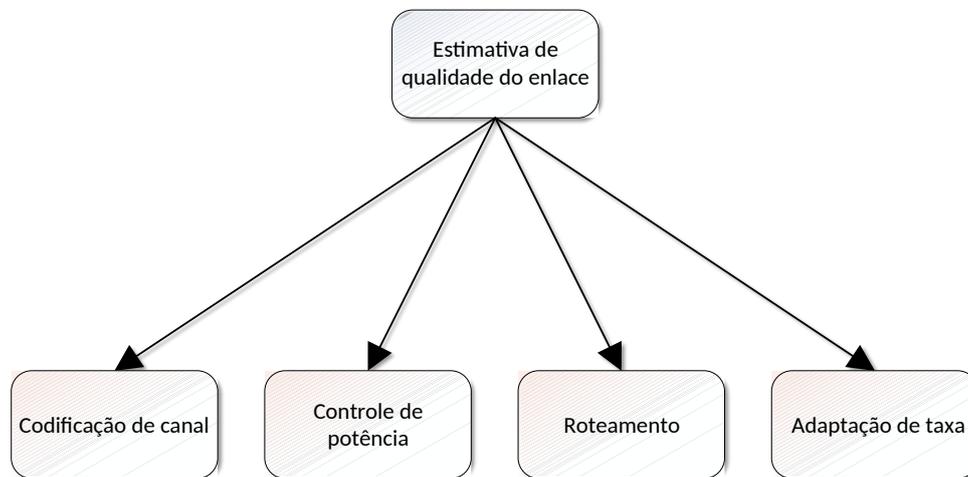


Figura 1.2. Aplicabilidade da estimativa de qualidade do enlace

O uso adequado do resultado de uma estimação precisa oferece melhorias da taxa de transferência da rede, da confiabilidade e da eficiência energética, propiciando o aumento da porcentagem de mensagens entregues com sucesso e a redução da quantidade de retransmissões em enlaces de baixa qualidade. Na Figura 1.2 são elencadas as principais técnicas influenciadas pelo uso de estimativas de qualidade do enlace.

Para alguns dispositivos M2M alimentados por baterias, além das problemáticas que permeiam a confiabilidade e eficiência, o consumo de energia também é uma preocupação e, à medida que se evitam retransmissões desnecessárias, aumenta-se a eficiência energética do sistema. As principais especificidades dos estimadores em comunicação M2M consistem em limitados recursos disponíveis (computacional e

energético) e em requisitos mais estritos como, por exemplo, a transmissão de dados periódica em ambiente ruidoso e com fonte de energia limitada.

1.2 Definições preliminares

Nesta seção são apresentadas breves definições de relevantes termos discutidos neste trabalho. O objetivo é evitar ambiguidades ou más interpretações com termos similares exibidos ao longo deste capítulo e no restante do texto.

Canal de comunicação: é o meio físico utilizado para transportar sinais elétricos/eletromagnéticos de um equipamento emissor a um receptor; pode estar associado a um meio de transmissão físico, como um fio de cobre, ou a uma conexão lógica sobre um meio multiplexado, ou um canal de rádio. Os sinais codificados representam dados ou mensagens.

Enlace de dados: é a conexão entre dispositivos de comunicação em dois ou mais locais, geralmente utilizando canais de comunicação para o transporte de dados.

Conectividade: é a capacidade ou propriedade que um equipamento microprocessado possui de estabelecer conexões com outros equipamentos.

Confiabilidade na transmissão: é a capacidade do sistema de realizar e manter a comunicação em condições preestabelecidas em um dado intervalo de tempo. De forma geral, significa que a conectividade deve ser assegurada, independentemente das condições do canal de comunicação.

Desempenho da comunicação: é a quantidade de entrega de dados úteis realizados com sucesso para um dispositivo e pode ser estimado em termos de precisão, eficiência e velocidade de transmissão em redes de comunicação.

Disponibilidade do dispositivo: é a probabilidade do equipamento estar presente e desempenhar suas funções sempre que necessário, sob determinadas condições. Por exemplo, ao longo de um ano, um sistema que apresenta 90% de disponibilidade possui um tempo de inatividade de 36,5 dias, ao passo que 99% representa 3,65 dias de inatividade.

Estimativa, estimador e estimação de canal de comunicação: **estimativa** é a avaliação ou cálculo aproximado da qualidade do canal de comunicação; **estimador** é o algoritmo ou protocolo que calcula a estimativa do canal; e

estimação do canal é o ato ou efeito de estimar, a partir de uma amostra, determinado valor que tenha máxima probabilidade de representar uma avaliação correta do canal.

1.3 Problemática

O desempenho de um sistema M2M é um fator crítico, especialmente para aquelas aplicações que têm requisitos de tempo real, e está diretamente ligado à confiabilidade de transmissão. Em ambiente com muitos ruídos e interferências, por exemplo, o desempenho do sistema pode ser reduzido em virtude das retransmissões necessárias para propiciar transmissão confiável.

Os problemas que serão investigados neste trabalho são a baixa confiabilidade e o limitado desempenho em comunicação M2M que opera em ambientes ruidosos. A relevância desses problemas é justificada pelo potencial de uso projetado dos dispositivos M2M e pela crescente concorrência pelo limitado espectro de frequência, e tem sua complexidade advinda da alta variabilidade da qualidade do enlace de comunicação em curtos períodos de tempo, dificultando a estabilidade e a precisão dos algoritmos de estimação. A transmissão infrequente, uma relevante característica dos sistemas de comunicação M2M [Lien et al., 2011], agrega uma dificuldade adicional na estimativa precisa da qualidade do enlace, visto que em um pequeno intervalo de tempo um canal pode ter sua qualidade deteriorada ou melhorada significativamente. Outro fator que gera dificuldade na estimação é o compromisso justo entre acurácia, responsividade, adaptabilidade, simplicidade e *overhead*, ou seja, o *tradeoff* para a operação do algoritmo estimador. Pode-se compreender o *tradeoff* como um desejado equilíbrio entre duas ou mais características desejáveis, mas simultaneamente incompatíveis. Na Figura 1.3, é apresentado graficamente esse equilíbrio que o algoritmo estimador precisa ponderar.

Vários trabalhos na literatura tentam tratar esses problemas, contudo, conforme será apresentado no capítulo de revisão bibliográfica, observa-se que ainda há lacunas quando se analisa os estimadores com processos de estimação envolvendo técnicas de aprendizagem, o que pode fornecer uma estimativa mais acurada. Ademais, nenhum dos estimadores de qualidade do enlace da literatura consideram *concept drift* em suas estratégias de decisão. No contexto deste trabalho, *Concept Drift* (CD) refere-se a um cenário de aprendizado de máquina em que o relacionamento entre os dados de entrada e a variável de destino sofre alterações ao longo do tempo, indicando que a hipótese do sistema de aprendizado de máquina possivelmente foi

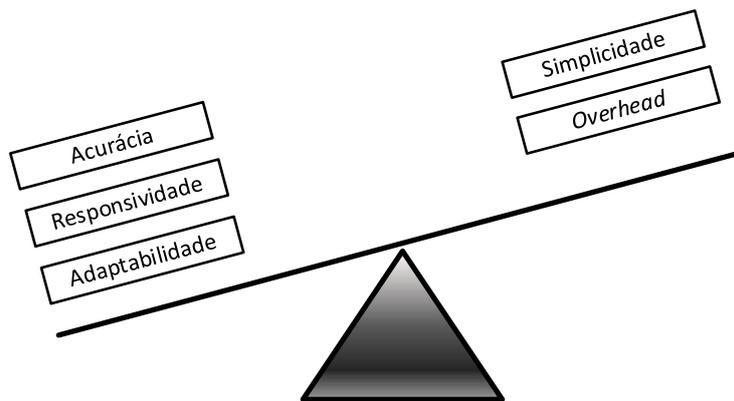


Figura 1.3. *Tradeoff* a ser considerado no processo de estimação de qualidade do enlace - adaptado de Lowrance & Lauf [2017]

alterada.

Resumindo a problemática, as principais questões que visamos responder neste trabalho são:

i) Dada a concorrência cada vez maior pelo limitado espectro de radiofrequência, como manter ou melhorar a confiabilidade da comunicação entre dispositivos máquina a máquina?

ii) Como o desempenho em comunicação M2M, com as especificidades inerentes, pode ser mantido ou favorecido em condições que propiciariam a degradação do sinal de transmissão?

iii) Diante da alta variabilidade da qualidade do enlace em curtos intervalos de tempo, é possível conceber um estimador de qualidade de enlace preciso, de rápida convergência e estável?

iv) As estratégias de aprendizado de máquina com uso de concept drift aplicadas para a estimação da qualidade do enlace podem fornecer resultados mais acurados?

1.4 Objetivos

O objetivo geral do trabalho visando a responder as questões de pesquisa elencadas anteriormente, consiste no estudo de métodos e técnicas de estimação de qualidade

do enlace em comunicação M2M, e na proposição de novos estimadores para melhoria da confiabilidade na transmissão entre dispositivos M2M.

Os objetivos específicos deste trabalho compreendem:

1. avaliação do efeito do processo de estimação em sistemas de comunicação M2M em condições adversas, como na presença de ruído eletromagnético e de outros distúrbios;
2. avaliação das contribuições de sistemas de aprendizagem de máquina (*machine learning*) para o processo de estimação da qualidade do enlace;
3. identificação dos efeitos que um sistema de adaptação e detecção de *Concept Drift* fornecem para o desempenho de algoritmos de aprendizagem de máquina no contexto dos estimadores de enlace.

1.5 Contribuições

Considerando as lacunas sobre a limitada confiabilidade na comunicação M2M e as dificuldades proporcionadas pela alta variabilidade do enlace, ambas descritas na Seção 1.3, e os objetivos descritos na (Seção 1.4), as principais contribuições que este trabalho fornece são:

- O projeto e desenvolvimento de um novo estimador de qualidade de enlace, multicamadas, híbrido, rápido, preciso e estável, denominado PRR²;
- O projeto e desenvolvimento de um segundo estimador de qualidade do enlace, multicamadas, com estratégias de decisão utilizando aprendizagem de máquina com detecção e tratamento de *concept drifts*, denominado LQM;
- Incorporação dos novos estimadores de enlace ao módulo do protocolo IEEE 802.15.4 existente no GNU Radio, ambiente de desenvolvimento para rádios definidos por *software*;
- Estudo e avaliação da confiabilidade e do desempenho em comunicação M2M em uma prova de conceito utilizando rádios USRP (*Universal Software Radio Peripheral*) reconfiguráveis com o ambiente GNU Radio.

Este trabalho se destaca dos anteriores por empregar dados de várias camadas com um sistema de aprendizado de máquina que utiliza o tratamento de *concept drift* para fornecer uma estimativa mais precisa. O estimador proposto considera

simultaneamente RSSI, SNR, PRR, latência e a taxa de pacotes transmitidos por pacotes recebidos (TPRP). As avaliações são conduzidas em experimentos reais usando um *testbed* de USRPs (modelo Ettus B200), o que leva a experimentos mais precisos, diretos e replicáveis do que simulações.

Lista de publicações

As pesquisas para este trabalho produziram as seguintes publicações científicas e submissão para periódico (em fase de réplica, aguardando segundo julgamento dos revisores):

Evento Qualis A1: da Silva, W. S.; Macedo, D. F.; Nogueira, M.; Nguyen, T. M. T. & Nogueira, J. M. S. (2017). A multilayer link quality estimator for reliable machine-to-machine communication. Em *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1--7. IEEE

Journal Qualis A1: Submissão (fase de réplica) do artigo “A Link Quality Estimator using Machine Learning with Concept Drift adaptation for M2M Communication” para o *Journal of Network and Computer Applications/Elsevier*

Capítulo de livro: Dutra Souza, A.; Marques, A. F.; Macedo, D. F.; Collins, D.; Júnior, G. M.; Cordeiro, J. R.; Marquez-Barja, J. M.; Kayo Coelho, K.; Pinto, L. R.; da Silva, L. A. et al. (2017). Experimental wireless networking research using software-defined radios. Em *Brazilian Symposium on Computer Networks and Distributed Systems-XXXV SBRC 2017*, pp. 1--49

Capítulo de livro: Silva, W. S.; Cordeiro, J. R. S.; Macedo, D. F.; Vieira, M. A.; Vieira, L. F. & Nogueira, J. M. S. (2015). Introdução a rádios definidos por software com aplicações em gnu radio. Em *Minicursos do XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos—SBRC*, volume 1, pp. 216--265

1.6 Organização do trabalho

O restante deste documento está estruturado conforme descrito a seguir. O **Capítulo 2** apresenta os fundamentos da pesquisa, tais como comunicação M2M, confiabilidade, controle de potência, estimadores de qualidade do enlace e aprendizagem

de máquina. O **Capítulo 3** apresenta a análise dos trabalhos relacionados à temática de pesquisa, expondo as principais soluções identificadas por outros autores e as lacunas existentes. O **Capítulo 4** detalha a proposta de estimador de qualidade de enlace denominado PRR². O **Capítulo 5** descreve o estimador desenvolvido que se utiliza de aprendizagem de máquina, denominado LQM. O **Capítulo 6** apresenta e discute os resultados experimentais dos estimadores propostos. O **Capítulo 7** discute as principais contribuições, resultados alcançados e limitações, apresenta a conclusão da tese, bem como as possibilidades de trabalhos futuros.

Capítulo 2

Fundamentos teóricos

A primeira parte deste capítulo compreende os principais conceitos de comunicação máquina a máquina, seus principais termos relacionados, e os seus principais requisitos. Em seguida, são descritos os fundamentos de controle de potência de transmissão e é feita uma discussão sobre os conceitos relacionados a estimadores de qualidade de enlace. Na segunda parte, são apresentadas as definições de séries temporais e os principais fundamentos sobre aprendizagem de máquina.

2.1 Comunicação Máquina a Máquina

A comunicação máquina a máquina (*Machine-to-Machine* - M2M), também denominada comunicação tipo máquina (*Machine-Type Communication* - MTC), é uma tecnologia de comunicação sem fio na qual os dispositivos inteligentes podem trocar informações e tomar decisões colaborativas de forma autônoma, sem ou com mínima interação humana [Anton-Haro & Dohler, 2014]. M2M constitui um paradigma em desenvolvimento que fornece conectividade ubíqua entre dispositivos como computadores, sistemas embarcados, sensores/atuadores inteligentes e dispositivos móveis.

O principal objetivo da comunicação M2M é aumentar o nível de automação de um determinado sistema, permitindo que os dispositivos possam trocar e compartilhar dados sem ou com limitada interação humana [Song et al., 2015]. Sua origem advém dos sistemas de supervisão e aquisição de dados (*Supervisory Control and Data Acquisition* - SCADA) [Stouffer & Falco, 2006], nos quais os sensores e outros dispositivos conectados por redes de transmissão de dados são utilizados com computadores para supervisionar e controlar os processos industriais, possibilitando um melhor desempenho na produção e uma redução nos custos operacionais [Kim et al., 2014].

O crescente interesse nas comunicações M2M também é justificado pelo grande potencial projetado nos próximos anos para o mercado M2M. Conforme relatado por Cisco [2019], o número de sistemas M2M móveis conectados crescerá quatro vezes até 2022, atingindo 3,9 bilhões em número, enquanto o tráfego M2M crescerá 8 vezes no mesmo período, atingindo 1,7 exabytes por mês em 2022.

As principais aplicações de M2M envolvem ambientes diversos, desde os domésticos até os industriais, entre elas cuidados com a saúde de pacientes, gestão eficiente de energia, entretenimento, monitoramento inteligente de consumos residenciais, controle e automação autônomos e precisos em processos fabris [Verma et al., 2016]. Ademais, as aplicações das redes de comunicação M2M não estão limitadas àquelas relacionadas com a indústria ou residência, podendo alcançar áreas como comunicação veicular, monitoramento ambiental, manutenção de equipamentos hospitalares, rastreamento logístico, segurança pública e cuidados com a saúde [Chen et al., 2014; Kartsakli et al., 2014; Verma et al., 2016; Xia et al., 2018].

A fim de prover a comunicação adequada para as diversas aplicações, uma estratégia relevante é adotar o gerenciamento de recursos de rádio (*Radio resource management* - RRM) [Xia et al., 2018], incluindo controle de acesso, alocação de recursos de rádio, gerenciamento de energia e adoção das mais recentes tecnologias que suportam comunicações M2M, como o 6LowPAN, ZigBee e CoAP. Para suportar as aplicações mencionadas, as redes M2M apresentam características específicas, tais como transmissão em massa e simultânea, demanda por confiabilidade na comunicação, agendamento e acesso prioritário ao meio, baixo consumo de energia, ausência ou limitada mobilidade, e baixa latência [Hussain et al., 2017]. Essas características serão melhor discutidas na Subseção 2.1.5.

O restante desta seção mostrará as principais diferenças entre os termos M2M e comunicações dispositivo a dispositivo (*Device-to-Device* - D2D), Internet das coisas (*Internet of Things* - IoT), redes de sensores sem fio (*Wireless Sensor Network* - WSN) e sistemas ciberfísicos (*Cyber-Physical Systems* - CPS). Em seguida, serão discutidos os principais modelos de sistemas e tráfego M2M, bem como as arquiteturas e padrões para as comunicações máquina a máquina.

2.1.1 Modelos de comunicação M2M

O modelo de comunicação em sistemas máquina a máquina é composto de três domínios interligados: de dispositivo M2M, de rede e de aplicação [Verma et al., 2016]. A estrutura é composta pelos elementos propostos pelo ETSI (*European Telecommunications Standards Institute*): o domínio de dispositivo M2M é formado

pela rede M2M e por dispositivos M2M; o domínio de rede é constituído por todos os tipos de infraestrutura de redes com ou sem fios (*x Digital Subscriber Line - xDSL*, fibra ótica, 3G *etc.*); e o domínio de aplicação consiste de destinatários que realizam o processamento final dos dados. A arquitetura geral desse modelo está ilustrada na Figura 2.1. Cada um dos domínios será discutido a seguir.

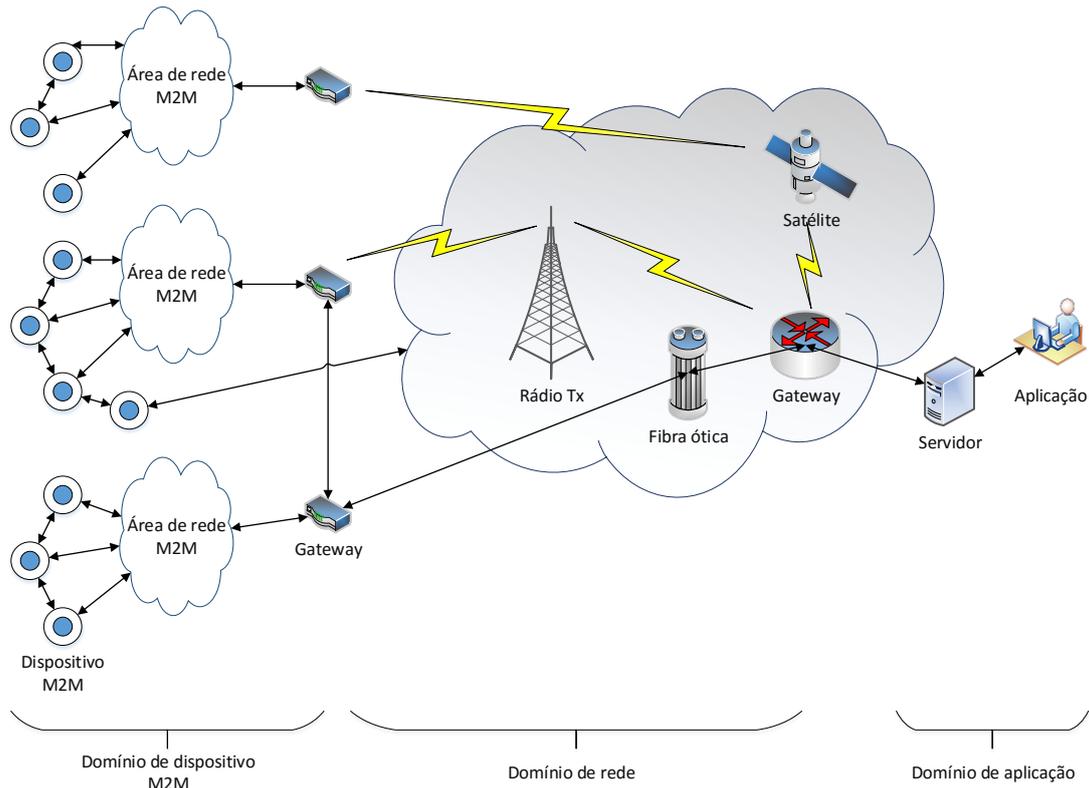


Figura 2.1. Modelo de sistema de comunicação M2M - adaptado de Verma et al. [2016]

2.1.1.1 Domínio de dispositivos

Este domínio é formado pelos diversos dispositivos M2M que colaborem entre si, tais como atuadores e sensores, e *gateways* (equipamentos que concentram as transmissões entre redes de comunicação). Esses dispositivos M2M coletam dados de diferentes partes da área de rede M2M e de forma colaborativa e constante, tomam decisões sobre a transmissão dos dados, com possibilidade de realizar, por exemplo, compressões de dados, controle da potência da transmissão e criptografia nas mensagens. Os *gateways* recebem os dados provenientes dos dispositivos M2M e os encaminham pela infraestrutura de rede até um servidor final do domínio de

aplicação, e também podem realizar trocas de mensagens diretas (*peer-to-peer*) com outros *gateways* do domínio de dispositivos M2M. Este domínio está representado na porção esquerda da Figura 2.1.

2.1.1.2 Domínio de rede

Este domínio atua como um serviço de transporte de dados entre os domínios de dispositivos e de aplicação. São utilizados diversos protocolos em uma ampla infraestrutura de redes cabeada e sem fio, desde redes telefônicas até comunicação por satélites. O principal objetivo é transportar de forma ágil e segura a informação coletada até o local de destino final dos dados. Este domínio está representado na porção central da Figura 2.1.

2.1.1.3 Domínio de aplicação

O domínio de aplicação consiste de servidores de armazenamento e de processamento de dados coletados no domínio de dispositivos, bem como de aplicações M2M. O servidor atua como um ponto de integração para armazenar toda a informação transmitida a partir do domínio de dispositivos M2M, possibilitando o acesso remoto aos dados monitorados. Sua função varia de acordo com a aplicação: em redes elétricas inteligentes, o centro de controle desempenha o papel de servidor, enquanto que em sistemas de monitoramento de saúde o servidor é um dispositivo M2M dedicado a essa função, ou estar localizado em uma central hospitalar, por exemplo. Este domínio está representado no segmento direito da Figura 2.1.

2.1.2 Tipos de comunicação

De acordo com Holler et al. [2014], os elementos de rede de um sistema M2M admitem comunicações da seguinte forma:

Dispositivo - dispositivo: neste tipo de comunicação o dispositivo M2M se conecta diretamente a um outro dispositivo M2M por meio de uma conexão direta. Este trabalho de tese concentra-se neste tipo de comunicação.

Dispositivo - *gateway*: neste tipo de comunicação o dispositivo M2M se conecta a um, ou mais, equipamento(s) *gateway* para que este possibilite a conexão com outros nós da rede.

Dispositivo - elementos do domínio de rede: neste tipo de comunicação, o dispositivo M2M pode se conectar diretamente a qualquer um dos elementos

integrantes do domínio de rede, desde que o dispositivo M2M possua a *interface* física e especificações lógicas necessárias para se comunicar com aquele equipamento de rede.

Dispositivo - servidor: neste tipo de comunicação o dispositivo M2M se conecta diretamente a um servidor de aplicação, demandando, para isso, que ambos trabalhem com o mesmo protocolo de comunicação. Este tipo de comunicação é pouco comum, mas pode ocorrer em situações controladas e ambientes fisicamente pequenos.

Gateway - gateway: neste tipo de comunicação um gateway se conecta diretamente a outro gateway para, dentre outros motivos, alcançar de forma mais rápida um determinado elemento do domínio de rede.

Elementos do domínio de rede - servidor: neste tipo de comunicação um elemento do domínio de rede se conecta diretamente ao um servidor de aplicação, possibilitando, por exemplo, a entrega e recebimento dos dados dos dispositivos M2M.

2.1.3 Tráfego M2M

Diferente de outros sistemas de comunicação, o tráfego nas comunicações M2M ocorre, predominantemente, no sentido *upstream* [Verma et al., 2016], e é classificado quanto ao tamanho dos pacotes, modo de transmissão e prioridade de acesso. Na Tabela 2.1 é apresentada a classificação do tráfego M2M para os principais tipos de aplicações.

A largura de banda indica a capacidade de transmissão da rede M2M ao qual o dispositivo M2M pertence e transmite as mensagens. Na Tabela 2.1, a largura de banda está representada por duas categorias: *pequena* e *grande*. A primeira refere-se a aplicações que requerem no máximo 100 kbps de dados, enquanto *grande* são aquelas com volume maior que 100 kbps.

O modo de transmissão representa qual a temporalidade da transmissão de dados, e está na Tabela 2.1 classificado como *periódico*, *aleatório* e *sob demanda*. O termo *periódico* neste contexto indica uma transmissão em intervalos de tempo regulares; *aleatório* é quando não há uma estimativa de quando ocorrerá a transmissão, pois pode depender de fatores externos; e *sob demanda* indica uma transmissão sempre que houver uma solicitação pelo lado cliente.

Prioridade de transmissão indica a premência relativa da transmissão, sendo classificada em *baixa*, *média* e *alta*.

Tabela 2.1. Requisitos de tráfego M2M conforme a aplicação - adaptado de Verma et al. [2016]

Aplicações M2M	Largura de banda	Modo de transmissão	Prioridade
<i>E-health</i>	Pequena	Periódico	Alta
Sistemas de navegação	Pequena	Aleatório	Média
Segurança pública e vigilância	Pequena	Sob demanda	Alta
Gerenciamento de energia doméstica	Grande	Periódico	Média
<i>Smart grid</i>	Grande	Periódico	Média
Aplicações elásticas (DTN)	Grande	Sob demanda	Alta
Telefonia (tempo real)	Pequena	Aleatório	Alta
Multimídia (atrasos adaptativos)	Grande	Sob demanda	Alta
Sistemas de logística em transporte	Pequena	Periódico	Alta
<i>Streaming</i> móvel	Grande	Sob demanda	Média
Medidores inteligentes	Pequena	Periódico	Alta
Serviços de emergência	Peq./grande	Aleatório	Alta

2.1.4 Arquiteturas e padrões M2M

A vasta aplicabilidade das comunicações M2M revela a necessidade de várias soluções técnicas que possam empregar sistemas M2M com o mínimo de falhas, expondo um novo desafio de interoperabilidade. Diferentes abordagens podem ser adotadas para a interconexão de dispositivos M2M de acordo com as aplicações, tais como as tecnologias com e sem fios, comunicações de curto ou de longo alcance, e soluções baseadas em padrões abertos de comunicação existentes ou tecnologias proprietárias.

A organização 3GPP (*3rd-generation partnership project*) e o ETSI (Instituto europeu de telecomunicações - *European Telecommunications Standards Institute*) concentram esforços para padronizar as comunicações M2M. Enquanto o 3GPP define o MTC (*Machine-Type Communication*) com foco nas comunicações, o ETSI define a arquitetura M2M com foco principal sobre as aplicações.

Várias esforços foram empregados para elaborar arquiteturas e padrões para servir como referência para os trabalhos em M2M, dentre os quais, destaca-se o projeto oneM2M. Lançado em 2012 como iniciativa do ETSI para garantir a implantação mais eficiente de sistemas de comunicações M2M e de IoT, o projeto de padronização **oneM2M** é um projeto de parceria global constituído por oito das principais organizações de desenvolvimento de padrões de tecnologia da informação do mundo: ARIB (*Association of Radio Industries and Businesses*, Japão), ATIS (*Alliance for Telecommunications Industry Solutions*, Estados Unidos), CCSA (*China Communications Standards Association*, China), ETSI (Europa), TIA (*Telecommunications Industry Association*, Estados Unidos), TSDSI (*Telecommunica-*

tions Standards Development Society, Índia), TTA (*Telecommunications Technology Association*, Coreia do Sul) e TTC (*Telecommunication Technology Committee*, Japão), e mais de 200 membros institucionais. O oneM2M agrega diversas áreas de negócios, incluindo telemática, transporte inteligente, saúde, utilidades, automação industrial e casas inteligentes [oneM2M, 2016]. Todos os padrões e especificações propostos pelo oneM2M, inclusive os rascunhos (*drafts*), estão disponíveis em ¹.

Inicialmente, o projeto oneM2M visa a planejar, aprovar e manter o conjunto necessário de especificações e relatórios técnicos e, finalmente, permitir a interoperabilidade em todo o ecossistema que envolve M2M e Internet das coisas. A arquitetura padronizada pelo oneM2M está em desenvolvimento, mas atualmente define uma **camada de serviço** de IoT, ou seja, um *software* independente de fornecedor que fornece um conjunto de funções necessárias para aplicativos de IoT. A camada de serviço oneM2M oferece funções independentes de caso de uso, como identificação de usuários e aplicativos, autenticação e autorização de usuários e aplicativos, criptografia de dados ponta a ponta e gerenciamento de dispositivo.

O 3GPP especifica três modos de comunicação no MTC: direta, indireta e por meio de serviços controlados (*Service Provider Controlled* - SPC). No modelo direto, a aplicação MTC comunica-se com o dispositivo MTC diretamente como uma aplicação *over-the-top* (OTT), ou seja, uma aplicação que está oferecendo um serviço através da Internet, mas que não é fornecido diretamente pelo servidor (*internet service provider*). No modelo indireto, a aplicação MTC faz uso de um servidor de MTC fornecido pelo operador; no modelo SPC, a aplicação MTC utiliza um servidor MTC fornecido por um provedor de serviço terceirizado. As seguintes características foram identificadas pelo 3GPP para MTC [Taleb & Kunz, 2012]:

- Baixa mobilidade: raro ou limitado movimento;
- Tempo controlado: entrega de dados apenas em intervalos de tempo predefinidos;
- Transmissões de dados limitadas: apenas pequenas quantidades de dados são transferidos;
- Transmissões pouco frequentes: longo período entre duas transmissões de dados;
- Segurança: conexão segura entre dispositivos e servidores MTC;

¹<http://onem2m.org/technical/published-drafts>

- Alarme de prioridade: geração de alarmes de prioridade após a ocorrência de um determinado evento.

2.1.5 Requisitos e características das comunicações M2M

Baseando-se majoritariamente nos trabalhos de Hussain et al. [2017], Xia et al. [2018], e Anton-Haro & Dohler [2014], são descritas adiante as principais características e os requisitos das comunicações M2M. É importante destacar que um determinado aspecto pode ser mais significativo que outro, de acordo com a aplicação M2M e o contexto ao qual está associado, conforme iremos discutir na subseção 2.1.1.

Transmissão em massa e simultânea de dados: é a transferência de um grande volume de dados utilizando um meio físico. Esta característica lida com tentativas de transmissões simultâneas para *gateways* da rede de acesso ou estação base a partir de um grande número de dispositivos M2M. Este atributo pode ser imprescindível para uma série de casos de uso, tais como saúde, segurança pública, acesso seguro e medição inteligente.

Confiabilidade: no contexto M2M, este termo significa o grau de fidelidade que uma conexão deve possuir para manter-se funcional, mesmo que as condições do enlace de comunicação estejam desfavoráveis. Esta característica é fundamental em situações de emergência em que a entrega dos dados é muito importante, como em serviços de cuidados de saúde e de pagamento remoto. A confiabilidade nas comunicações M2M pode ser dividida em duas classes: a disponibilidade de dispositivos e a confiabilidade da transmissão [Misic & Misic, 2014]. A primeira se refere à capacidade de estar pronto sempre que necessário, e tem como uma possível abordagem de solução a adoção de redundância no sistema. A segunda classe, confiabilidade da transmissão, está associada à transmissão e à qualidade da comunicação sem fio e é afetada, por exemplo, pelas variações no comportamento do canal sem fio e por protocolos de operação nas camadas física e MAC. Este trabalho foca no segundo entendimento de confiabilidade. Uma definição mais formal associa confiabilidade à probabilidade de que um dos serviços da rede funcione adequadamente em um intervalo de tempo $(0, t]$ [Silva et al., 2012].

Baixo consumo de energia: a inviabilidade de recarga ou troca da bateria, na maioria dos dispositivos M2M, torna a energia um fator limitante para a correta operação do sistema. Esta característica é fundamental principalmente em

dispositivos com restrições severas de energia, que devem ser ativados somente sob demanda, e após completar a tarefa entram em estado de inatividade.

Tráfego em rajadas: trata-se das transmissões que demandam grande largura de banda em um curto período de tempo. A maioria dos dados transmitidos por dispositivos M2M é por rajada [Anton-Haro & Dohler, 2014].

Reduzida mobilidade: diversos cenários de aplicação M2M envolvem dispositivos fixos ou de limitada mobilidade (e.g. medições industriais e sensores de batimento cardíaco).

Segurança: os dispositivos M2M implantados no campo ficam expostos a ataques de rede, *hardware* e *software*. Tais vulnerabilidades podem comprometer as credenciais de acesso e as configurações de transmissão e monitoramento. Para evitar ocorrências dessa natureza, os dispositivos M2M devem ser capazes de detectar eventos incomuns, tais como localização de dispositivo modificado e propiciar níveis adequados de autenticação para dispositivos M2M.

Baixa latência: refere-se tanto à baixa latência de transmissão, ou seja, a diferença de tempo entre o pacote enviado e a sua respectiva confirmação, quanto à latência de transmissão de dados. Esta característica é fundamental em diversos cenários de emergência (e.g., cuidados de saúde e vigilância), tornando-se um importante fator que difere as comunicações M2M dos demais tipos de redes sem fio, como as redes de sensores e redes *ad hoc*.

2.1.6 Termos relacionados: D2D, IoT, WSN e CPS

As relações entre M2M, D2D, IoT, WSNs e CPS podem parecer nebulosas em um primeiro contato, mas há particularidades em cada sistema que podem ser discutidas. Esta subseção objetiva esclarecer as principais diferenças e similaridades entre esses sistemas.

Comunicação dispositivo a dispositivo (*Device-to-Device* - D2D): é um tipo de comunicação M2M na qual os dispositivos estão próximos entre si e se comunicam sem a mediação de uma estação rádio base, mas geralmente requerem interação humana para as requisições ou envio de dados. D2D foi proposto por Lin & Hsu [2000] e a tecnologia mais utilizada para comunicação é MTC (*Machine-type Communication*) sobre a rede de telefonia celular e pode operar em dois modos: dentro do espectro de frequência da rede celular

(*inband*) e fora do espectro do LTE (*outband*) [Asadi et al., 2014]. O primeiro modo reutiliza a frequência LTE disponível (ou LTE-A), e o modo *outband* realiza uma mudança para uma frequência de espectro não licenciado, como a faixa de 2,4 GHz. Essas características fazem com que D2D possibilite expandir a cobertura de uma rede de comunicação M2M.

Internet das coisas (*Internet of Things - IoT*): é uma infraestrutura de rede que liga objetos físicos e virtuais através da captura de dados e comunicação a uma plataforma que possibilita a execução de uma aplicação [Gubbi et al., 2013]. Essa infraestrutura de comunicação inclui a Internet, outras redes existentes e em desenvolvimento. A tecnologia IoT requer identificação específica dos objetos e capacidade de interconexão como princípios para o desenvolvimento de serviços cooperativos independentes. Algumas vezes ambos os termos (M2M e IoT) são utilizados de forma intercambiável; no entanto, a característica mais importante de IoT é a informação que os objetos conectados podem fornecer, e como essas informações podem ser combinadas e apresentadas por uma aplicação. Uma definição proposta por Wan et al. [2013] indica que as tecnologias M2M são algumas das principais ferramentas para a Internet das Coisas [Bhat & Dohler, 2015].

Redes de sensores sem fio - RSSF (*Wireless Sensor Network - WSN*):

são compostas de nós sensores autônomos espacialmente distribuídos para monitorar condições físicas e/ou ambientais [Clare et al., 1999]. Nós sensores são dispositivos autônomos com capacidades de sensoriamento, processamento e comunicação. Estas redes possuem grande aplicabilidade em locais de difícil acesso ou perigosos, tais como as áreas militares, industriais e de aviação, dentre outras. Essa dificuldade de acesso muitas vezes é um impedimento para a substituição das baterias que alimentam os nós sensores, de forma que o consumo de energia torna-se um fator crítico. O uso de redes de sensores possibilita ampliar a capacidade de obtenção de informações em sistemas M2M, uma vez que elas podem aumentar a capilaridade da cobertura dos sistemas. As RSSF distinguem-se das redes M2M principalmente pela restrita cobertura geográfica, ao passo que as redes M2M podem ter cobertura em escala global.

Sistema ciberfísico (*Cyber-Physical System - CPS*): refere-se a uma nova geração de sistemas com capacidade de interação com seres humanos [Baheti & Gill, 2011]. Um CPS geralmente é composto por elementos computacionais

colaborativos com o intuito de controlar entidades físicas. Os elementos de computação de um CPS comunicam-se com sensores que monitoram indicadores virtuais e físicos, e atuadores, que modificam o ambiente virtual e físico em que são executados. CPSs utilizam-se de nós sensores para conectarem toda a “inteligência” distribuída no ambiente para obter um conhecimento mais profundo do mesmo, possibilitando uma atuação mais precisa [Baheti & Gill, 2011]. Em um contexto físico, atuadores agem e modificam o ambiente em que estão inseridos. Em um contexto virtual, CPSs podem ser usados para coletar dados das atividades virtuais de usuários, como seu envolvimento em redes sociais, *blogs* ou sites de comércio eletrônico. Com as capacidades de tomada de decisão e controle autônomos, sistemas M2M podem ser atualizados para sistemas ciberfísicos. Sob a ótica da arquitetura da Internet das coisas, CPS foi proposto como uma evolução do M2M, com a introdução de operações mais inteligentes e interativas [Wan et al., 2013].

2.2 Controle de Potência de Transmissão

O controle de potência de transmissão (*Transmit Power Control* - TPC) é um mecanismo utilizado em sistemas de comunicação sem fio com o propósito de reduzir a potência de um transmissor de rádio ao mínimo necessário para manter a ligação com bom desempenho e qualidade, de acordo com os requisitos do sistema [Gao et al., 2014]. Um eficiente esquema de controle de potência também contribui para aumentar a vida útil da bateria de dispositivos, fator determinante em alguns equipamentos, como os que estão situados em local de difícil acesso para manutenções.

Também chamado controle dinâmico de potência (*Dynamic Power Control* - DPC), os mecanismos de controle de potência podem propiciar, em decorrência dos ajustes na potência, redução nas interferências em outros dispositivos, melhoria da confiabilidade da comunicação e redução do consumo energético. Nos sistemas de comunicação sem fios, o TPC torna-se um dos principais componentes para permitir transmissões de dados com baixas interferências e aumentar a capacidade do sistema [Cotuk et al., 2014].

A maioria das estratégias de controle de potência (TPC) consistentemente não ajusta ao máximo a potência de transmissão por um tempo significativo, mas sim define um nível ideal para a comunicação em cada dispositivo em um determinado período [Raza et al., 2012], [Lin et al., 2016]. Se fosse adotada uma potência máxima por todo o período de transmissão de dados, o consumo de energia seria elevado

mesmo em situações em que não seria necessário potência máxima. Assim, um algoritmo para executar TPC deve considerar em suas decisões a eficiência energética do sistema.

Para sustentar essas decisões, o equipamento monitora métricas do sinal e/ou da camada de rede a fim de determinar se o nível de potência de transmissão precisa ser alterado para assegurar a comunicação com eficiência energética. Uma vez que o nível ideal é alcançado e definido, o mecanismo permanece avaliando se são necessárias novas mudanças.

Uma forma de avaliar a confiabilidade de um enlace sem fio com relação à potência de transmissão é relacionar a probabilidade de falhas na comunicação à probabilidade da relação sinal-ruído (*Signal-to-Noise Ratio* - SNR) ser menor que um limite previamente estabelecido, indicando que a qualidade do enlace pode ser insuficiente para atender os requisitos de transmissão. Seja γ o símbolo que denota o SNR e P_t a potência do sinal, $\gamma = (P_t/N_o)g$, N_o é o ruído e g é o ganho do enlace, tem-se que a probabilidade de falha na comunicação, $P_o(\Gamma_o)$, é dada por:

$$P_o(\Gamma_o) = Pr \left\{ \frac{P_t}{N_o}g \leq \Gamma_o \right\} \quad (2.1)$$

em que Γ_o representa o limite mínimo de SNR aceitável. Outros métodos de avaliação da confiabilidade podem utilizar métricas baseadas em erros de bit, como a taxa de erros de bit (*Bit Error Rate* - BER).

Sob a ótica do controle de potência, o desempenho em comunicações M2M pode estar relacionado com a taxa de entrega do sistema durante uma transmissão, ou com a capacidade de transmissão em um certo momento, ou ainda com a eficiência energética durante as comunicações. Desse modo, o desempenho pode ser utilizado como um parâmetro, em conjunto com a confiabilidade, para uma correta compreensão do comportamento do sistema [Pan & Hu, 2014].

Para a decisão do nível de potência de transmissão, muitos algoritmos TPC utilizam-se dos valores obtidos pelos estimadores de qualidade do enlace (*Link Quality Estimators* - LQE), tais como RSSI e PRR. A próxima seção apresenta os conceitos relacionados aos LQEs.

2.3 Estimadores de qualidade do enlace

Estimadores de qualidade de enlace (*Link quality estimators* - LQE) são funções ou algoritmos desenvolvidos para estimar a qualidade do canal de comunicação entre dois ou mais dispositivos [Lowrance & Lauf, 2017]. Podem utilizar-se de métricas

estatísticas provenientes da contagem de pacotes ou de indicadores físicos, como a potência instantânea do sinal em uma certa frequência [Lowrance & Lauf, 2017].

Estimadores precisos são fundamentais para protocolos sem fio eficientes, e fornecem uma caracterização estatística dos enlaces usando teoria de estimação. A estimativa geralmente é obtida através da avaliação de uma ou mais métricas de rede, que avaliam o enlace, dentro de uma janela de estimação w [Baccour et al., 2015].

O processo de estimação envolve basicamente três etapas: monitoramento, medição do enlace e avaliação da métrica [Baccour et al., 2012]. Tais etapas serão melhor discutidas na Subseção 2.3.1. Os principais requisitos dos estimadores são eficiência energética, precisão, reatividade, estabilidade e limitado *overshoot*, e serão detalhados na Subseção 2.3.2. Aliada a esses requisitos, há a natureza dinâmica das redes sem fio, em que é alta a variabilidade da qualidade da comunicação, tornando desafiante a obtenção de um estimador rápido e que satisfaça a maioria desses requisitos.

2.3.1 Etapas

O processo de estimar a qualidade de um enlace pode ser dividido em três etapas: monitoramento do enlace, medições do enlace, e avaliação da métrica. De acordo com Baccour et al. [2012], o **monitoramento do enlace** analisa as condições do canal dentro de uma janela de estimação w , e pode ser classificado em três tipos: ativo, passivo e híbrido. No monitoramento ativo o nó monitora o canal de acesso aos seus vizinhos através do envio de sinais de sondagem. No monitoramento passivo o nó escuta o meio sem induzir a sobrecarga de comunicação extra, ou seja, não envia nenhum sinal ou pacote. O mecanismo híbrido combina técnicas de monitoramento do ativo e passivo, atribuindo pesos diferentes de acordo com as estratégias adotadas.

A **medição do enlace** é o processo que recupera informações úteis a partir do sinal ou dos pacotes recebidos e/ou enviados, dependendo da técnica utilizada. Dados provenientes de pacotes recebidos são utilizados para computar a qualidade do enlace do lado receptor (*receiver-side* LQE), enquanto os dados obtidos dos pacotes enviados podem ser utilizados para calcular a qualidade do enlace do lado transmissor (*sender-side* LQE).

A etapa de **avaliação da métrica** é para selecionar qual a métrica mais representativa para indicar a qualidade do enlace. Em geral, esta etapa é a fase crucial no processo de estimação: de acordo com a técnica adotada, antes da computação dos valores, pode ser utilizada suavização dos dados, ou outras técnicas como filtragem

e decisão por aprendizado de máquina. Na Figura 2.2 estão ilustradas essas etapas, apresentando a sequência de processamento de tráfego para a obtenção do valor de saída do LQE.

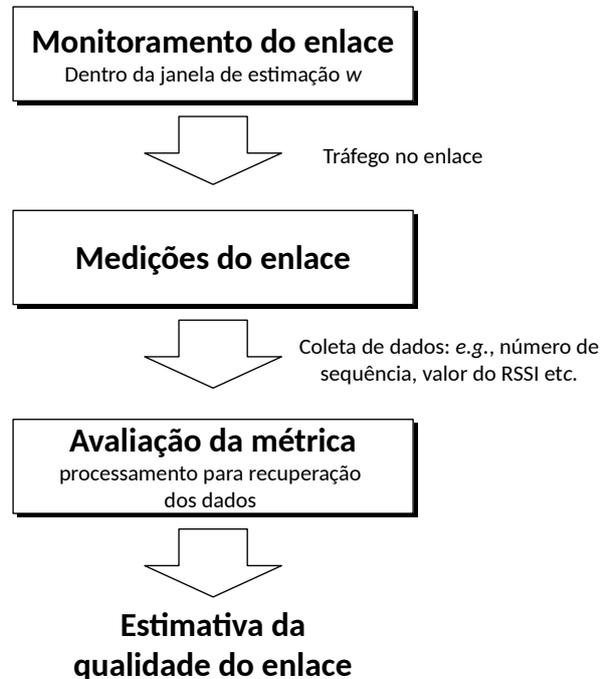


Figura 2.2. Etapas para estimativa de qualidade do enlace – adaptado de Baccour et al. [2012]

As etapas e o mecanismo de operação dos estimadores de qualidade do enlace também podem ser compreendidos como um sistema de controle. Em termos gerais, um sistema de controle é uma interconexão de componentes relacionados a fim de comandar, controlar ou ajustar a si ou outro sistema [Åström & Murray, 2010]. A Figura 2.3 apresenta o diagrama de blocos de um sistema de controle com retroalimentação, no qual a saída medida é considerada pelo controlador, juntamente com a referência. Os elementos que compõem o sistema estão brevemente descritos a seguir [Hellerstein et al., 2004]:

- Referência: é o valor esperado das saídas das medições, ou de suas transformações;
- Erro: é a diferença entre a saída medida e a referência;

- Controlador: determina o valor necessário da entrada do sistema para que a saída alcance o valor de referência, considerando as entradas atuais e valores de erros anteriores;
- Distúrbios: quaisquer alterações que afetem o meio no qual o sistema está operando;
- Ruídos: são fatores externos ou internos que modificam o valor de saída medido;
- Sistema alvo: sistema a ser controlado;
- Saída medida: trata-se de uma característica específica do sistema alvo;
- Transdutor: é o elemento responsável por comparar a saída medida com o valor de referência.

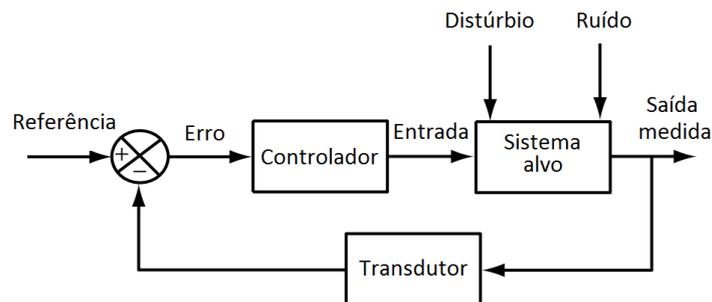


Figura 2.3. Diagrama de blocos de um sistema de controle - adaptado de Hellerstein et al. [2004]

2.3.2 Requisitos

A maioria das propriedades e requisitos dos estimadores de qualidade do enlace pode ser considerada como pertencente ao conjunto de requisitos dos sistemas de controle, dentre os quais se destacam: estabilidade, acurácia, tempo para alcançar a estabilidade e baixo *overshoot*. Tais requisitos podem ser compreendidos como:

Estabilidade: refere-se à capacidade de tolerar oscilações provenientes das entradas e distúrbios;

Acurácia: trata-se da proximidade entre o valor de saída medido na estabilidade e o valor de referência, de forma que quanto mais próximo for, mais acurado será o sistema;

Tempo de alcance da estabilidade (*settling time*): é o intervalo de tempo necessário para o sistema convergir a um valor próximo ao esperado mantendo-se estável;

Overshoot e undershoot: é o valor inicial que a variável observada ultrapassa o valor final, no caso do *overshooting*, ou que fica aquém do valor final, se *undershooting*, usualmente expressos como uma porcentagem do valor final.

A Figura 2.4 apresenta um exemplo de uma curva do comportamento de um sistema de controle de processos.

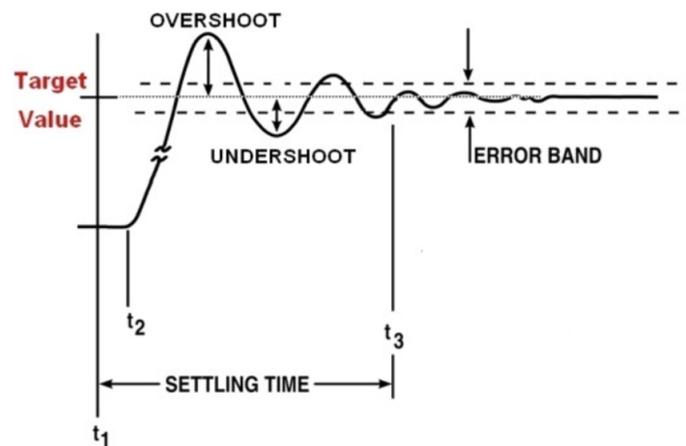


Figura 2.4. Curva de resposta de um sistema de controle de processos - Hellerstein et al. [2004]

Além dos requisitos provenientes dos sistemas de controle, temos ainda requisitos específicos dos estimadores de qualidade de enlace: eficiência energética, acurácia, reatividade e estabilidade, conforme detalhados a seguir:

Eficiência energética: é a capacidade do estimador de enlace fornecer estimativas com o mínimo de erros e com o mínimo de energia empregada;

Acurácia: refere-se à capacidade da técnica de LQE caracterizar corretamente o estado do enlace, ou seja, capturar o comportamento do canal; possui grande impacto na eficácia de outros protocolos de rede [Baccour et al., 2011];

Reatividade: está relacionada à capacidade do estimador de reagir a variações significativas na qualidade do canal;

Estabilidade: refere-se à capacidade de tolerar oscilações curtas na qualidade do enlace [Baccour et al., 2012]. Dessa forma, percebe-se que estabilidade e a reatividade são ortogonais, e um compromisso deve ser alcançado entre essas exigências.

2.3.3 Classificação dos LQEs

Diversas classificações para os estimadores de qualidade do enlace são apresentadas na literatura: Renner et al. [2011] realiza a classificação dos estimadores em baseados em métricas físicas e baseados em métricas lógicas. Baccour et al. [2012] classifica os estimadores em baseados em *hardware* e em baseados em *software*. Em razão de propiciar uma classificação mais precisa, com pouca margem para erros na classificação, neste trabalho será adotada a classificação apresentada em Heinzer [2011], que subdivide os estimadores em baseados na potência do sinal e baseados em estatísticas de pacotes.

Uma classificação taxonômica dos estimadores está ilustrada na Figura 2.5 e resume as classificações quanto às métricas utilizadas e aos requisitos. Os estimadores PRR² e LQM propostos neste trabalho são híbridos, e se encaixariam na taxonomia utilizando métricas baseadas em potência do sinal e em estatísticas de pacotes.

2.3.4 Métricas baseadas em potência do sinal

As métricas baseadas em potência do sinal são obtidas diretamente dos circuitos por meio da observação do sinal eletromagnético. Sua principal característica é que dispensam computação adicional para obtenção do valor final. Uma significativa desvantagem dos estimadores que se baseiam unicamente em métricas de potência do sinal é que são, geralmente, menos precisos do que os estimadores baseados em estatísticas de pacotes, devido à sua inépcia em obter os níveis de potência em uma posição distante do equipamento no qual o estimador está atuando [Baccour et al., 2015]. As principais métricas nessa categoria são o RSSI, LQI e SNR.

Received Signal Strength Indicator - RSSI é uma medida da potência de radiofrequência de um canal, incluindo as outras transmissões de dados na mesma frequência, ou radiações de fundo, ou ruídos térmicos. A métrica RSSI também pode ser utilizada para o *Clear Channel Assessment (CCA)* [Engstrom & Gray, 2008]. CCA é uma técnica utilizada para identificar uma possível transmissão em

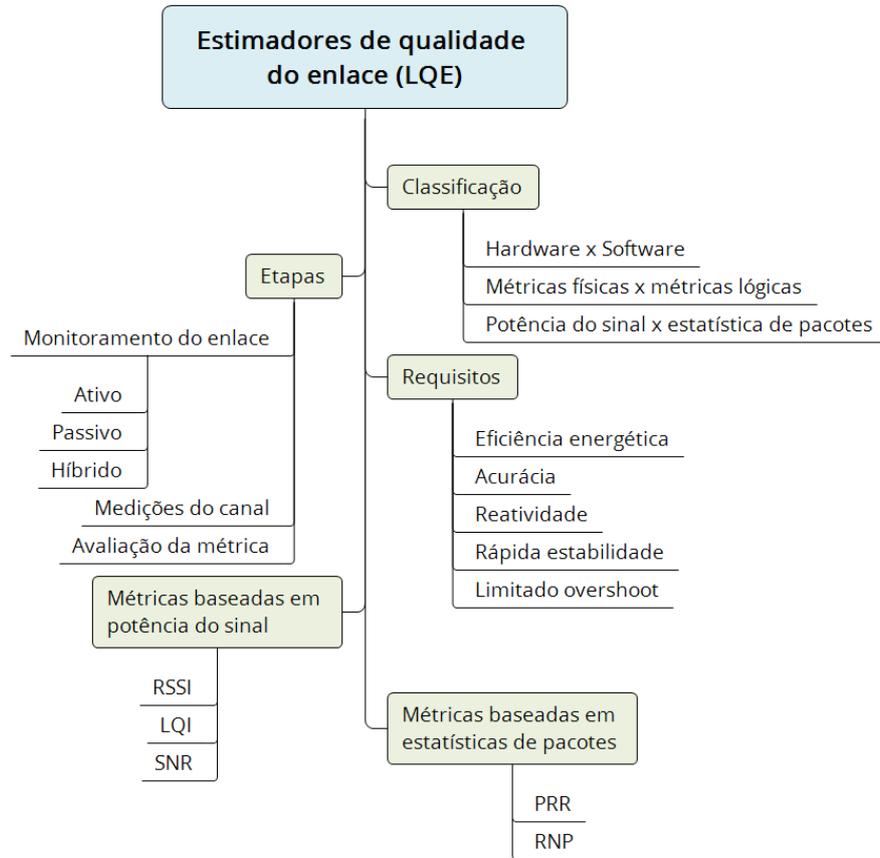


Figura 2.5. Classificação dos estimadores de qualidade de enlace

um tempo instantâneo por meio da detecção de níveis de energia acima dos limites estabelecidos para uma transmissão naquele canal [King & Roedig, 2018].

Link Quality Indicator - LQI é um indicador de qualidade do enlace proposto no padrão IEEE 802.15 [IEEE, 2013] e é transmitido como um campo na estrutura de cada pacote, e sua implementação é específica de acordo com o fabricante. A única exigência da referida norma é que ele deve ser calculado utilizando-se uma janela de no mínimo oito observações. Em redes multissaltos seus valores podem ser modificados a cada etapa à medida que a mensagem se propaga por meio dos equipamentos.

Signal-to-noise ratio - SNR compara o nível de um sinal desejado para o nível de ruído de fundo e pode ser expresso como $SNR = \frac{P_{sinal}}{P_{ruído}}$. SNR é frequentemente utilizado para estimar a taxa de erro de bits (*Bit error rate* - BER), que pode ser extrapolada para a taxa de erros de pacotes recebidos.

Os principais métodos LQE neste grupo são *Kalman-filter-based* LQE

(KLE) [Senel et al., 2007], *LQE-node* [Gomes et al., 2017b], *Rep* [Rojas & Decotignie, 2018] e o de *Qin et al.* [Qin et al., 2018]. Esses estimadores serão descritos em detalhes na Seção 3.

2.3.5 Métricas baseadas em estatísticas de pacotes

As métricas que são determinadas a partir da computação de indicadores estatísticos de pacotes, sejam recebidos ou enviados, são chamadas de baseadas em estatísticas de pacotes. A independência de um *hardware* específico é a principal vantagem dos estimadores que se utilizam de métricas baseadas em estatísticas de pacotes. As principais métricas deste grupo são PRR e RNP.

Packet Reception Ratio - PRR representa a taxa de pacotes recebidos com sucesso durante uma janela de observação, e pode ser expressa como $PRR = \frac{ReceivedPkt}{SentPkt}$, onde *ReceivedPkt* é o número de pacotes recebidos com êxito e *SentPkt* é o número de pacotes enviados. É amplamente utilizada em protocolos de roteamento e também como métrica imparcial para avaliar a precisão dos estimadores baseados em potência do sinal. O tamanho da janela de observação define a eficiência do PRR.

Required Number of Packet retransmissions - RNP calcula a quantidade média de retransmissões necessárias de pacotes medida durante uma janela de observação w e pode ser expressa conforme a Equação (2.2):

$$RNP = \frac{PktTx}{PktRx} - 1, \quad (2.2)$$

em que $PktTx$ é o número médio de pacotes transmitidos e retransmitidos durante uma janela de estimação e $PktRx$ é o número de pacotes recebidos com sucesso. Subtrai-se um para eliminar o primeiro pacote transmitido.

Os principais métodos LQE neste grupo são *Window Mean with Exponentially Weighted Moving Average (WMEWMA)* [Woo & Culler, 2003], *Expected Transmission Count (ETX)* [De Couto et al., 2005], *Link Inefficiency (LI)* [LaI et al., 2003], *Four-bit* [Fonseca et al., 2007], *L-NT e L-ETX*, [Zhang et al., 2010], *E2E-DDR* [Sun et al., 2017b], *Clustering link quality estimation (CLQE)* [Ferng & Abdullah, 2019], *4C (Foresee)* [Liu & Cerpa, 2014], *Anfis LQE (A-LQE)* [Vallejo et al., 2014], *WNN-LQE* [Sun et al., 2017a] e *Random Vector Functional Link based Link Quality Prediction - RVFL-LQP* [Xue et al., 2020]. Esses estimadores serão descritos em detalhes na Seção 3.

Os dados fornecidos pelas métricas que são utilizadas pelos estimadores de enlace são, em uma análise estatística, classificados como séries temporais. Em razão disso, os estimadores modernos realizam tratamentos e operações comuns às séries

temporais, bem como se utilizam de algoritmos de aprendizagem de máquina para fornecer um resultado mais acurado. A seguir serão apresentados os fundamentos teóricos básicos sobre Séries temporais e Aprendizagem de máquina.

2.4 Séries temporais

Séries temporais são um conjunto de observações sequenciais $(x_1, x_2, x_3, \dots, x_t)$ ao longo do tempo t . É importante destacar que a variável tempo pode ser substituída por outra variável, como espaço ou profundidade [Morettin & Toloi, 2006].

A dependência temporal (*i.e.*, memória) entre observações em tempos diferentes pode ser caracterizada pela avaliação de uma função de auto-correlação (ACF) $\rho(k)$. A ACF mede a similaridade entre uma série X_t e sua versão deslocada X_{t+k} :

$$\rho(k) = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2}, \quad (2.3)$$

$$\mu = E[X_t], \quad (2.4)$$

$$\sigma^2 = E[(X_t - \mu)^2], \quad (2.5)$$

onde E , k , μ and σ representam, respectivamente, valor esperado, atraso (*lag*), média e desvio padrão.

Uma série temporal é dita contínua no tempo quando as observações podem ser medidas em qualquer ponto no tempo. Definindo um conjunto $T = \{t : t_1 < t < t_2\}$, a série temporal pode ser denotada por $\{x(t) : t \in \mathbb{R}\}$. Diz-se que uma série temporal é discreta quando observações são feitas em momentos específicos, geralmente com espaçamento iguais. Definindo o conjunto $T = \{t_1, \dots, t_n\}$, a série temporal será denotada por $\{x_t : t \in T\}$. Para simplificar, pode-se fazer $T = \{1, 2, 3, \dots, n\}$.

2.4.1 Técnicas de predição e suavização

Predição ou previsão é um problema bastante significativo para diversas ciências. Os métodos variam de acordo com o problema e os dados. Devido à dificuldade em modelar problemas reais, várias técnicas surgiram para simplificar e auxiliar as séries temporais de previsão que descrevem tais problemas. As principais técnicas de previsão são a modelagem estatística, o sistema de modelagem inteligente ou uma modelagem híbrida usando ambas as técnicas [Morettin & Toloi, 2006]. Pre-

dição é o processo de antecipação dos valores de dados futuros com base em dados passados e presentes. Suavização são técnicas simples de previsão que assumem que os valores extremos presentes em uma série representam a aleatoriedade e, desse modo, a suavização desses extremos permite identificar um padrão básico [Morettin & Toloï, 2006].

No contexto dos estimadores de qualidade de enlace, é importante utilizar técnicas de predição e suavização para suavizar flutuações curtas e/ou destacar as tendências de longo prazo presentes na série temporal. As técnicas clássicas de predição e suavização de séries temporais utilizadas neste trabalho são Média Móvel (*Moving Average* - MA) e Suavização Exponencial Única (*Single Exponential Smoothing* - SES). Este trabalho adota as técnicas previamente citadas e também incorpora técnicas de normalização, computação por valores mínimos, e estratégias inteligentes de predição, tais como aquelas relacionadas com aprendizagem de máquina (*machine learning*) e estratégias de detecção e tratamento de *concept drifts*, discutidas em seção específica adiante.

2.4.1.1 *Moving Average* - MA

O método de previsão Média Móvel usa como previsão para um determinado ponto futuro a média de observações passadas. Este método fornece uma previsão de curto prazo, o que o torna uma escolha interessante quando os dados não apresentam tendência ou sazonalidade. As médias móveis são apresentadas como simples, centradas ou ponderadas. Modelos de médias simples y_t adotados neste estudo podem ser expressos como

$$y_t = \hat{X}_t = \frac{x_{t-1} + x_{t-2} + \dots + x_{t-n}}{n}, \quad (2.6)$$

onde n significa o tamanho da janela do tempo necessário para determinar a média e x_t o valor previsto.

O termo média móvel é usado porque, à medida que a próxima observação se torna disponível, a média das observações é recalculada, incluindo essa observação no conjunto de observações e negligenciando a observação mais antiga [Morettin & de Castro Toloï, 1981]. Este método só é recomendado para prever séries estacionárias; caso contrário, a precisão da previsão será prejudicada, devido aos pesos atribuídos às observações de n serem todos iguais e nenhum peso ser dado às observações anteriores nesse período.

2.4.1.2 *Single Exponential Smoothing* - SES

O método conhecido como suavização exponencial simples (SES) é semelhante à Média Móvel por extrair das observações das séries temporais o comportamento aleatório por suavização de dados históricos. No entanto, a inovação introduzida pela SES vem do fato deste método atribuir pesos diferentes para cada observação na série. Enquanto no MA as observações são usadas para encontrar o valor futuro e contribuem em igual proporção para o cálculo desta previsão, no SES as informações mais recentes são destacadas pela aplicação de um fator que determina sua importância [Wheelwright et al., 1973].

O SES é útil como um método para suavizar (ou filtrar) variações aleatórias e tem as seguintes propriedades: (i) a redução do peso é atualizada em um dado antigo; (ii) é extremamente fácil de calcular; e (iii) uma quantidade mínima de dados é necessária para computar [Makridakis et al., 2008]. O argumento para o tratamento diferenciado de algumas observações da série temporal baseia-se no pressuposto de que as últimas observações contêm mais informações sobre o futuro e, portanto, são mais relevantes para a previsão. Segundo Morettin & Toloí [2006] o método SES pode ser descrito pela seguinte equação:

$$\hat{Z}_t = a\hat{Z}_t + (1 - a)\hat{Z}_{t-1}, Z_0 = Z_1, t = 1, \dots, N \quad (2.7)$$

onde a é a constante de suavização e Z_t o valor exponencialmente suavizado. Se a equação for resolvida recursivamente, pode-se observar que os valores antigos de Z_t exercem menos influência na predição \hat{Z}_t .

2.5 Aprendizagem de máquina

Aprendizagem de máquina (do inglês *Machine Learning*) é um conjunto de métodos computacionais de extração de conhecimento a partir de uma base de dados, capaz de detectar padrões que podem ser utilizados para predição de dados futuros ou na tomada de decisões [Murphy, 2013]. Alguns desses métodos podem, por exemplo, detectar automaticamente não-linearidades nas relações entre variáveis independentes e dependentes e podem, automaticamente, identificar as interações entre as variáveis e seus fatores [Buskirk, 2018].

Na Figura 2.6 observa-se o modelo de programação tradicional, que gera uma saída de acordo com dados fornecidos a um programa desenvolvido por um especialista. Na Figura 2.7 tem-se o modelo geral dos sistemas de aprendizagem de

máquina, que diante de um conjunto de dados e de um conjunto de saída, fornece uma hipótese.

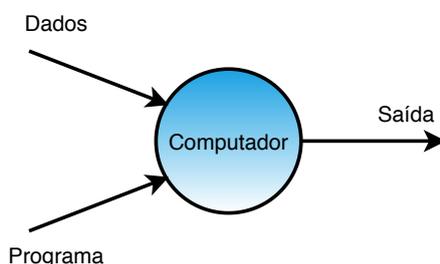


Figura 2.6. Programação tradicional – adaptado de [Lee, 2019]

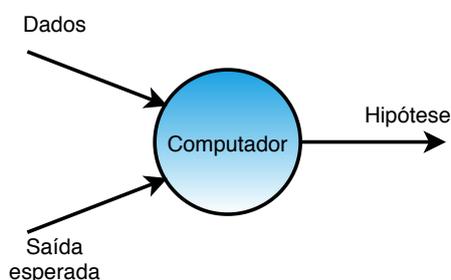


Figura 2.7. Aprendizagem de máquina – adaptado de [Lee, 2019]

A aprendizagem de máquina é, geralmente, classificada em três modos: aprendizado supervisionado, não supervisionado e por reforço. No **aprendizado supervisionado**, um algoritmo constrói um modelo matemático a partir de um conjunto de dados que contém as entradas (*inputs*) e as saídas desejadas (*targets*), fornecidas por um especialista (ou professor). Os algoritmos de **aprendizado não-supervisionados** não recebem saídas desejadas (*targets*) para os dados de entrada, devendo encontrar, sem uma supervisão, um padrão nas entradas fornecidas. O aprendizado não supervisionado pode ser um objetivo em si mesmo (descobrir novos padrões nos dados) ou um meio para atingir um fim. O **aprendizado por reforço** lida com o problema de aprender a ação apropriada ou sequência de ações a serem tomadas para uma dada situação, a fim de maximizar o retorno. Para isso, é fornecido ao algoritmo um *feedback*, na medida em que o sistema de aprendizagem continua a execução e os estados do sistema são explorados. As representações gráficas de tais modelos podem ser observadas, respectivamente, nas Figuras 2.8, 2.9 e 2.10, que apresentam os agrupamentos corretos de dados que antes estavam desorganizados ou sem classificação definida.

É observado na Figura 2.8 que o algoritmo de aprendizagem supervisionada realiza esse processo de agrupamento recebendo rótulos de um especialista ou supervisor para, em conjunto com os dados de entrada, diferenciar os grupos. No processo ilustrado na Figura 2.9 tem-se que o algoritmo não supervisionado promove a classificação dos dados em grupos de afinidade sem que um rótulo seja previamente fornecido. Na Figura 2.10 o processo de agrupamento é realizado recebendo uma recompensa por cada ação correta realizada.

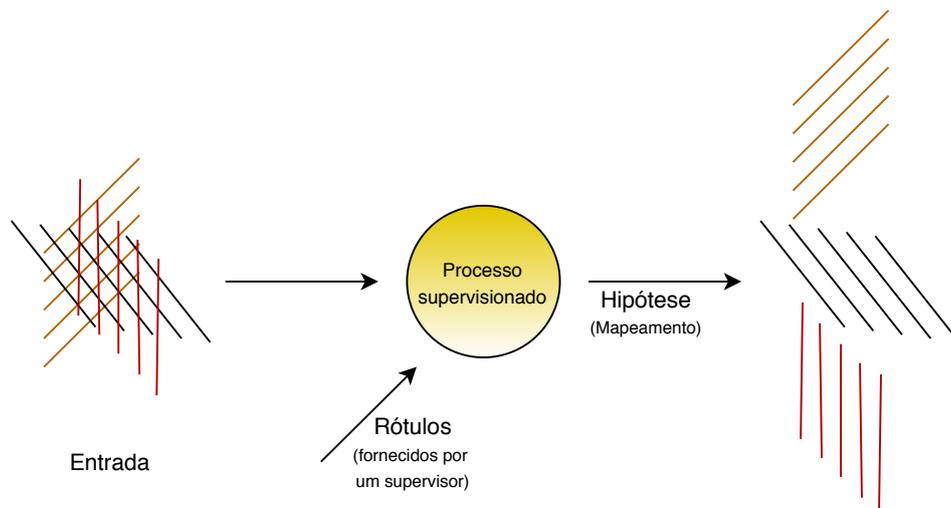


Figura 2.8. Modelo de processo supervisionado de aprendizagem de máquina – adaptado de [Jones, 2017]

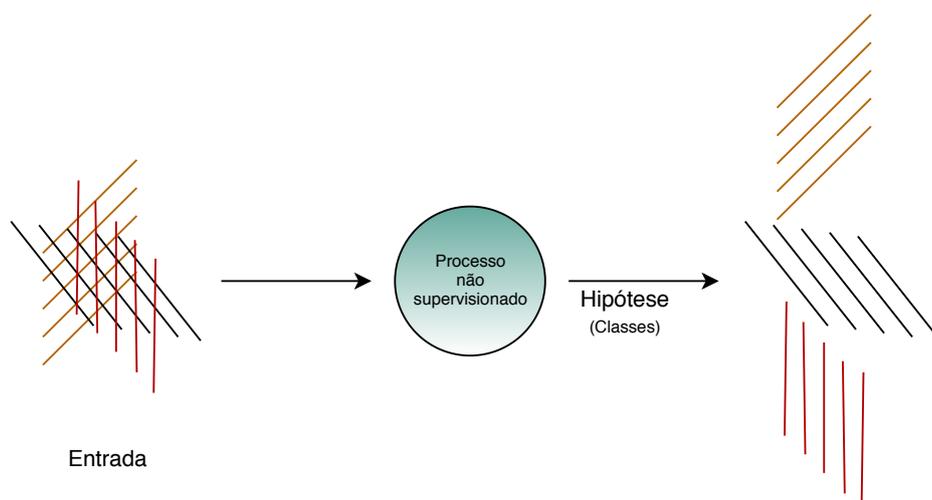


Figura 2.9. Modelo de processo não supervisionado de aprendizagem de máquina – adaptado de [Jones, 2017]

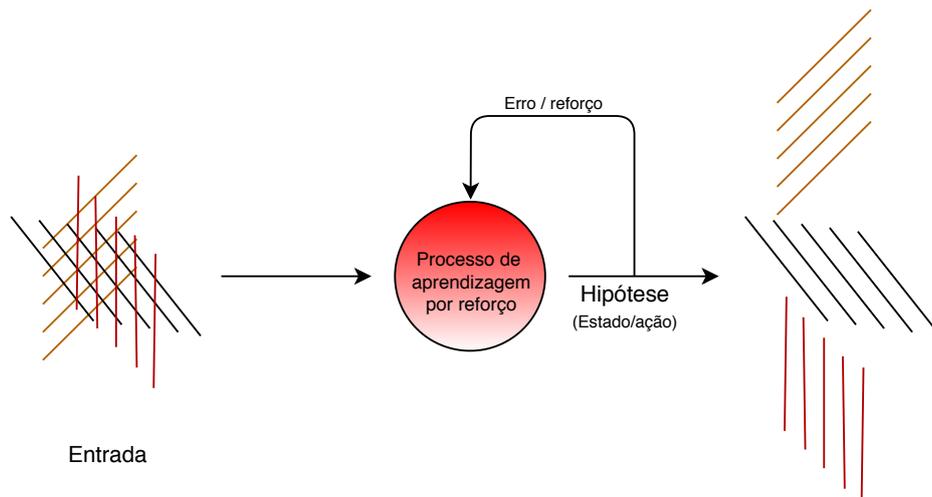


Figura 2.10. Modelo de processo de aprendizagem por reforço – adaptado de [Jones, 2017]

Uma categorização dos modelos de aprendizado de máquina supervisionado está relacionada às saídas desejadas: as aplicações em que o rótulo de destino é um número finito de categorias distintas, são conhecidas como tarefas de classificação. Casos em que o rótulo de destino é composto de uma ou mais variáveis contínuas são conhecidos como regressão. Em sistemas de aprendizado de máquina não supervisionado, as tarefas mais comuns são de agrupamento (*clustering*) e redução de dimensionalidade.

2.5.1 Aprendizagem supervisionada

Em um sistema de aprendizagem de máquina supervisionada, tem-se dados de entrada (x) e uma variável de saída (Y), e utiliza-se um algoritmo para aprender a função de mapeamento da entrada para gerar a hipótese.

$$Y = f(x) \quad (2.8)$$

O objetivo é aproximar a função de mapeamento de tal modo que, quando novos dados de entrada (x) forem apresentados, seja possível prever as variáveis de saída (Y) para esses dados. Esse processo é chamado de aprendizado supervisionado porque a base de dados para o algoritmo que aprende com o conjunto de dados de treinamento pode ser comparada com um supervisionamento de um professor, que acompanha o processo de aprendizado. Diante das respostas corretas, o algoritmo iterativamente faz previsões sobre os dados de treinamento e é corrigido pelo super-

visor. A aprendizagem é interrompida quando o algoritmo atinge um nível aceitável de desempenho ou quando se encerram os dados para treinamento, de acordo com parâmetros estabelecidos.

Na aprendizagem supervisionada é usado um conjunto de dados rotulados, fornecendo um significado para os dados. Usando o rótulo, dados que não estejam rotulados podem ser previstos para obter um novo rótulo [Lee, 2019]. Em outras palavras, o processo de aprendizado supervisionado ocorre a partir da apresentação de um conjunto de exemplos de treinamento ao algoritmo de aprendizagem, cabendo a ele gerar uma hipótese h , que é uma aproximação da verdadeira função f , que por sua vez é desconhecida. De posse de um novo exemplo não rotulado x_i , h deve ser capaz de prever a sua saída $f(x_i)$ [Sanchez, 2003].

A maioria dos problemas de aprendizagem supervisionada pode ser agrupada da seguinte forma:

Classificação: um problema é do tipo classificação quando a variável de saída é uma categoria, como “grande” - “pequeno” ou “tumor benigno” - “tumor maligno”, ou “ocupado” - “não ocupado”. Com a classificação, objetiva-se identificar a qual conjunto de categorias (subpopulações) uma nova observação pertence, com base em um conjunto de dados de treinamento contendo observações. Na Figura 2.11 tem-se um exemplo de um modelo para classificar tumores em benignos ou malignos, respectivamente sinalizados com os símbolos O e X, de acordo com a idade do paciente e o tamanho do tumor. O modelo forneceu uma divisão linear que melhor separa os dois tipos de tumores amostrados.

Regressão: um problema é do tipo regressão quando a variável de saída é um valor contínuo ou real, como “preço”, “potência” ou “peso”. Na Figura 2.12 está ilustrado um exemplo de modelo de regressão para identificar o valor de uma casa de acordo com o seu tamanho, em m^2 .

2.5.2 *Concept drift*

Os dados de uma série temporal se modificam de tal modo que suas características ao longo do tempo também podem ser modificadas, dificultando ou impossibilitando um modelo preditivo trabalhando com aqueles dados manter a precisão na sua estimativa. No contexto de aprendizado de máquina, *Concept Drift* (CD) refere-se a um cenário de aprendizado de máquina em que o relacionamento entre os dados de entrada e a variável de destino muda ao longo do tempo, ou seja, quando a hipótese

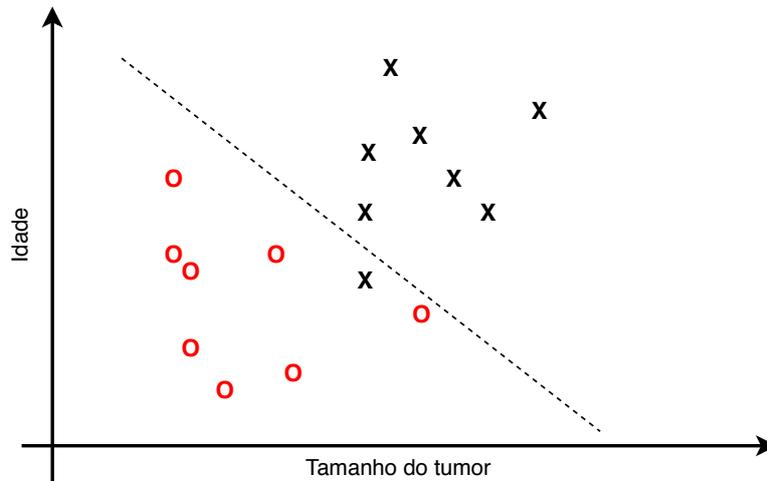


Figura 2.11. Exemplo de classificação – adaptado de [Lee, 2019]

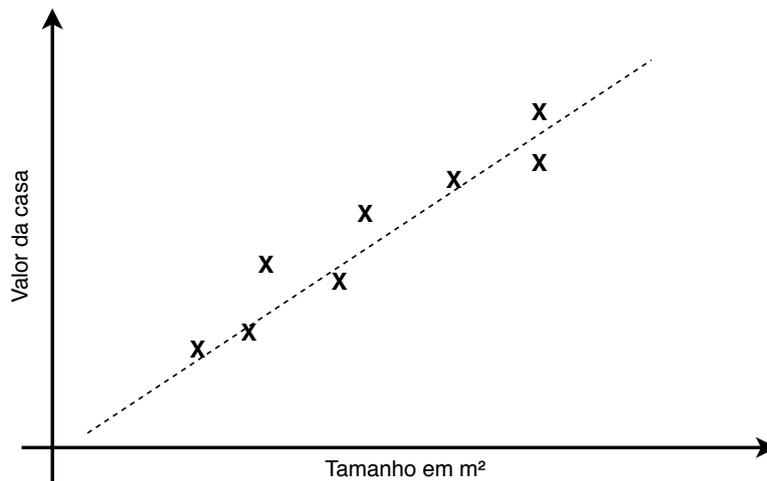


Figura 2.12. Exemplo de regressão – adaptado de [Lee, 2019]

possivelmente foi alterada. O CD está relacionado a alterações na distribuição de saída (*ie*, variável alvo), enquanto a distribuição de entrada pode permanecer inalterada [Gama et al., 2014]. Se um *Concept Drift* for identificado mas não for tratado, a acurácia da previsão do algoritmo poderá diminuir com o tempo.

O *Concept Drift* pode ser detectado usando os algoritmos ADWIN e CUSUM. O algoritmo ADWIN (*ADaptive sliding WINdow*) [Nishida & Yamauchi, 2007] é um detector de alterações que examina todos os cortes possíveis de uma determinada janela e calcula os tamanhos ideais das sub-janelas de acordo com a taxa de alteração entre duas sub-janelas. A estatística do teste é a diferença das duas médias da

amostra. Quando a distinção ultrapassa um limite com um intervalo de confiança predefinido, um corte ideal é identificado. Muitos métodos de identificação de CD na literatura são derivados ou combinados com o ADWIN, como os Bifet & Gavalda [2009]; Bifet et al. [2009]; Gomes et al. [2017a].

O algoritmo de soma cumulativa (*Cumulative Sum Algorithm* - CUSUM) [Manly & Mackenzie, 2000] é um teste sem memória e sua precisão depende da seleção na escolha dos parâmetros α e λ , de acordo com as Equações 2.9 e 2.10. CUSUM emite um alarme quando a média dos dados de entrada é significativamente diferente de zero, detectando aumentos e diminuições significativos.

$$g_0 = 0 \quad (2.9)$$

$$g_t = \max(0; g_{t-1} + (x_t - \alpha)) \quad (2.10)$$

onde g representa a variável observada, x_t representa as observações e t representa o tempo. A regra geral de decisão para detectar incrementos é

$$\text{se } g_t > \lambda \text{ então alarma e } g_t = 0$$

Uma variante do CUSUM é o teste de Page-Hinkley [Page, 1954]. Esta última é uma técnica de análise sequencial normalmente utilizada para detecção de alterações em processamento de sinais.

Nos algoritmos de aprendizado supervisionado, de acordo com Lu et al. [2018], existem muitas estratégias para tratar *Concept Drift* nos modelos de aprendizado: a primeira possibilidade é periodicamente reajustar o modelo. Alguns modelos de aprendizado de máquina podem ser atualizados sem a necessidade de um retreino completo; outra ação é atualizar periodicamente a base de treinamento do modelo de aprendizado; outra abordagem é utilizar algoritmos de aprendizado por reforço. Este trabalho adota a estratégia de retreino da aprendizagem de máquina quando é detectado *Concept Drift*.

2.6 Conclusão

A maioria dos atributos de confiabilidade e desempenho em comunicações M2M são de alta sensibilidade a variações na qualidade do enlace, tais como interferências, ruídos e congestionamentos. Atrasos e falhas na transmissão também impactam diretamente a confiabilidade das redes sem fio. Como resultado dessas características,

são explicados os principais mecanismos que podem ser aprimorados, tais como o gerenciamento da potência de transmissão e o estimador de qualidade de enlace.

Estimadores de qualidade de enlace precisos são requisitos fundamentais para alcançar comunicações sem fio confiáveis e eficientes. Conforme relatado por De Couto et al. [2005], um método LQE impreciso ou lento pode causar uma desaceleração superior a 200 % na vazão (*throughput*) da rede, principalmente devido ao algoritmo de roteamento operar de uma forma ineficiente. LQE precisos também são cruciais para diversos mecanismos na comunicação de dados, tais como a determinação da melhor rota, o controle eficiente da técnica de código de canal (*channel code*), adaptação da taxa de transmissão (*rate adaptation*) e os algoritmos de controle da potência de transmissão. Como discutido nas seções anteriores, esses algoritmos podem impactar diretamente a capacidade do sistema e a eficiência energética.

Embora muitas pesquisas proponham novos estimadores de qualidade do enlace, não é trivial obter um método LQE preciso e robusto, essencialmente considerando o comportamento dinâmico da rede sem fio.

Capítulo 3

Trabalhos relacionados

Os métodos estimadores de qualidade do enlace empregam uma ou mais métricas dentre aquelas descritas no Capítulo 2. Neste capítulo serão apresentados e discutidos os principais LQEs organizados pela quantidade de métricas utilizadas como entrada, *i.e.*, LQEs baseados em uma única métrica ou uma única camada de rede, e naqueles baseados em informações de múltiplas camadas da rede, ambos sem uso de processos aprendizagem de máquina. Uma terceira categoria, denominada LQEs baseados em processos de aprendizagem, é utilizada neste capítulo para agregar as técnicas que se utilizam de combinações de métricas e de estratégias de aprendizagem em seu processo de estimação. Na Figura 3.1, é apresentada uma taxonomia geral dos estimadores de acordo com a organização adotada neste trabalho.

3.1 LQEs baseados em única métrica ou camada de rede

O método LQE *Window Mean with Exponentially Weighted Moving Average* (**WMEWMA**) utiliza um filtro EWMA (*Exponential Weighted Moving Average*) para combinar estimativas de PRR previamente calculadas e evitar flutuações transitórias do PRR. No entanto, WMEWMA pode ser impreciso em algumas situações porque não indica as repentinas alterações na qualidade do enlace com suficiente agilidade. Esse estimador é de camada única, utilizando métricas de apenas uma camada da pilha de protocolos [Woo et al., 2003].

O *Expected Transmission Count* (**ETX**) é um estimador localizado no lado do receptor e possui uma estratégia semelhante à métrica RNP (*Required Number of Packet Retransmissions*). Ele estima a qualidade da transmissão nas direções

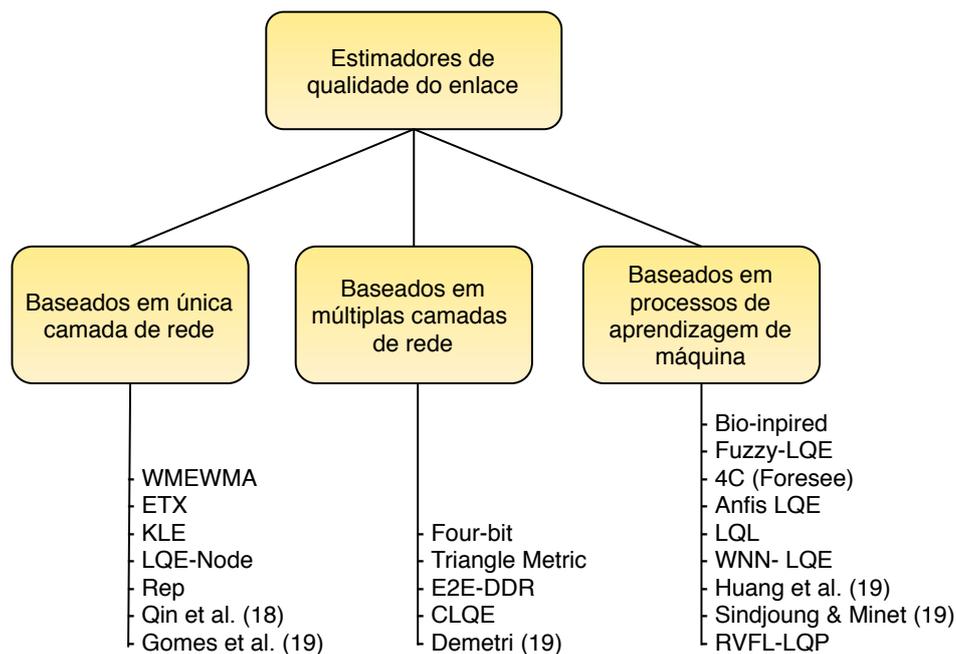


Figura 3.1. Taxonomia dos estimadores discutidos nesta seção

de enlace *uplink* e *downlink*, separadamente. A combinação dos dois valores estimados fornece uma estimativa da qualidade do link bidirecional. O WMEWMA e o ETX não lidam com variações abruptas da qualidade do link, porque não monitoram métricas baseadas em *hardware*, como o RSSI [Couto et al., 2003].

O estimador ***Kalman-filter-based LQE (KLE)*** usa um esquema com filtro Kalman para estimar o PRR a partir do RSSI. Os autores avaliaram sua proposta utilizando sensores com transceptores CC2420 executando o protocolo ZigBee [Senel et al., 2007].

Gomes et al. [2017b] apresentaram o **LQE-node** (*Link Quality Estimation-node*) para avaliar a qualidade do enlace usando métricas baseadas em *hardware* (por exemplo, RSSI), considerando a interferência de múltiplos caminhos. Embora o LQE-node e o KLE utilizem a métrica RSSI, uma desvantagem é que ambos os métodos não podem fornecer uma estimativa precisa em caso de perda de pacotes, pois eles empregam apenas métricas de *hardware*.

O estimador **Rep** utiliza um esquema de amostragem para extrair a qualidade do enlace a partir de amostras do preâmbulo de MACs de protocolos de baixa potência de transmissão. Este estimador utiliza LQI e RSSI em sua abordagem e usou 10 nós TelosB para avaliação [Rojas & Decotignie, 2018].

Qin et al. [2018] apresentam um estimador baseado em RSSI que usa um fil-

tro Kalman para ambientes industriais. Ele se destaca por utilizar variações na taxa de entrega como parâmetro. Foram utilizados USPRs para validar o experimento.

Gomes et al. [2019] propuseram um novo estimador para redes de sensores sem fio externa, que usa nós dedicados para estimar a qualidade do enlace. Para isso, o estimador proposto combina até três métricas: a análise do RSSI dos pacotes recebidos, os valores do RSSI obtidos em períodos sem recepção de pacotes e o número de pacotes duplicados recebidos (alta correlação com o PRR); no entanto, a análise experimental prática utiliza uma versão simplificada do estimador, com a implementação apenas do RSSI dos pacotes recebidos. Foram utilizados microcontroladores PIC18LF4620 conectados a transceptores MRF24J40MA.

3.2 LQEs baseados em informações de múltiplas camadas de rede

O LQE **Four-bit (4B)** utiliza quatro bits de informação: um da camada física, representando a qualidade do canal durante uma transmissão de pacotes; um da camada de enlace, indicando se os pacotes foram entregues e reconhecidos; e dois bits da camada de rede, que identificam quais *links* são mais valiosos para o desempenho da camada superior. Um protótipo mostrou melhorias significativas no custo e na taxa de entrega quando comparado a um *Link Quality Indicator (LQI)* de múltiplos saltos. Os autores empregaram o TinyOS 2 e um banco de testes baseado nos modelos TelosB para avaliar o desempenho de sua proposta [Fonseca et al., 2007].

O estimador **Triangle Metric LQE** combina as informações de PRR, LQI e SNR. O receptor calcula a média dos valores LQI e SNR em uma janela de estimativa de 10 pacotes e uma segunda função considera as informações de PRR [Boano et al., 2010].

O estimador **End-to-End Data Delivery Reliability (E2E-DDR)** utiliza as seguintes métricas como entrada: distância entre nós, ruído de fundo, condições ambientais e estados de *hardware*. No estimador E2E-DDR, uma estrutura captura a função de mapeamento entre o PRR, o ruído de fundo e o RSSI. Para verificar e validar o modelo, os autores escolheram uma rede de distribuição de energia como exemplo de aplicação de automação industrial [Sun et al., 2017b].

O estimador **Clustering link quality estimation (CLQE)** considera as informações de mobilidade e a estimativa da qualidade dos enlaces das redes veiculares urbanas (*Vehicular ad hoc networks - Vanets*). Ele lida com velocidade relativa média, SNR e PRR. O esquema proposto é avaliado através do simulador NS-3 com um

modelo realista de mobilidade de veículos gerado pelo simulador de tráfego SUMO (*Simulation of Urban MObility*) [Feng & Abdullah, 2019].

Demetri et al. [2019] propõem uma abordagem híbrida para estimar a qualidade do enlace nas redes Lora. Ela utiliza PRR e imagens geográficas da cobertura do solo para calcular a estimativa. Os *gateways* da rede *The Things Network* e o dispositivo Dragino LoRa shield v1.3 foram utilizados incorporando o chip de rádio RF96.

3.3 LQEs baseados em processos de aprendizagem de máquina

Os estimadores a seguir são baseados em estatísticas de pacotes e utilizam processos de aprendizagem de máquina. Uma deficiência comum desses métodos é que eles não têm a capacidade de considerar novos dados que se desviam dos padrões anteriores, como um *concept drift*, que pode resultar em estimativas imprecisas ou desatualizadas. Os estimadores que adotam o aprendizado de máquina convencional possuem dois componentes principais: treinamento/aprendizado e previsão. Um estimador que lida com *concept drift* (CD) requer outros três componentes: detecção do CD, compreensão e adaptação ao CD.

O estimador **Bio-inspired LQE** [Caleffi & Paura, 2009] é um estimador baseado em redes neurais. O processo de avaliação usou um simulador considerando uma rede *mesh* 802.11b [Cacciapuoti et al., 2014].

Baccour et al. [2010] apresentam o **Fuzzy-LQE (F-LQE)**, um estimador de qualidade de enlace baseado em lógica *fuzzy*. O F-LQE utiliza funções de associação lineares para calcular a estimativa de qualidade com base em PRR, assimetria de *link*, estabilidade e SNR. O estimador reduz o número de retransmissões de pacotes em até 32% [Baccour et al., 2015]. Os autores empregaram o dispositivo TelosB motes para avaliar o desempenho do estimador [Polastre et al., 2005].

Liu & Cerpa [2014] introduziram **4C (Foresee)**. O 4C aplica a previsão da qualidade do enlace junto com uma abordagem de aprendizado de máquina para prever a qualidade temporal de curto prazo do enlace com as informações da camada física e do PRR. A saída é a probabilidade de sucesso de entrega do próximo pacote. As experiências foram conduzidas no *testbed* Motelab.

O estimador **Anfis LQE (A-LQE)** [Vallejo et al., 2014] foi projetado para sistemas de comunicação sem fio corporal. Ele é baseado no RSSI e usa o sistema de inferência *Neuro-Fuzzy* para prever a qualidade do enlace. O A-LQE ajusta a po-

tência de transmissão dos nós distribuídos na superfície do corpo. Os experimentos utilizaram a plataforma de sensores sem fio Shimmer (*Sensing Health with Intelligence, Modularity, Mobility and Experimental Reusability*) [Burns et al., 2010].

O estimador **Link Quality Learning (LQL)** [Di Caro et al., 2013] utiliza PRR e *Locally Weighted Projection Regression (LWPR)* como métricas. O estimador LQL foi avaliado por simulação (NS-3), considerando cenários de rede móvel *ad hoc* e usando a plataforma de teste INDRIYA [Doddavenkatappa et al., 2011].

O estimador **WNN-LQE** utiliza um algoritmo baseado em rede neural Wavelet para determinar a estimativa de qualidade do enlace em redes de sensores sem fio de Smart Grid; no entanto, depois de coletar as métricas de *hardware*, os dados são carregados no *software* Matlab e o algoritmo é executado no modo de simulação [Sun et al., 2017a].

No trabalho de **Huang et al. [2019]**, os autores combinam métricas baseadas em *hardware* e *software* aplicando lógica *fuzzy* para estimativa da qualidade do enlace. Simulações de redes móveis de sensores sem fio mostraram que o método proposto apresentou taxas de precisão mais altas em comparação com PRR, LQI, SNR, Triangle Metric, WMEWMA e CSI.

O estimador proposto por **Sindjoug & Minet [2019]** prediz a qualidade do *link* de uma rede TSCH (*Time-Slotted Channel Hopping*) utilizando o aprendizado de máquina aplicado em duas métricas: RSSI e PDR (*Packet Delivery Ratio*). Eles adotaram o modelo de Regressão Linear (LR), o Linear Support Vector Machine (Linear SVM), SVM e RandomForest. O método foi avaliado no *testbed* Grenoble utilizando nós M3. As análises do Machine Learning foram realizadas offline e os autores planejam usar a estratégia no futuro em sistemas de IoT *online*.

O estimador **RVFL-LQP** (Random Vector Functional Link based Link Quality Prediction) utiliza uma rede neural RVFL não iterativa projetada para obter uma previsão rápida e precisa de duas partes da sequência SNR decomposta. A rede RVFL é um caso especial de redes neurais *feed-forward* de camada oculta única. A análise experimental utilizou nós transceptores CC2530 e obstruções aleatórias entre nós WSNs [Xue et al., 2020].

A Tabela 3.1 apresenta uma comparação sintética dos principais LQEs discutidos, de acordo com o local de atuação, sentido e avaliação utilizada. O local de atuação se refere ao lado que o estimador atua de fato, ou seja, onde os dados e algoritmo são processados para gerar a estimativa. O campo direção especifica sobre qual caminho, se *uplink* ou *downlink*, o estimador trabalha. O campo avaliação resume ao leitor quais foram as estratégias utilizadas pelos autores para promover a avaliação do seu estimador proposto.

Tabela 3.1. Tabela comparativa de LQEs

Estimador	Entradas	Local de atuação	Direção	Avaliação
WMEWMA [Woo et al., 2003]	PRR	Receptor	downlink	Simulação em Matlab
ETX [Couto et al., 2003]	RNP	Receptor	bidirecional	Testbed com placas Cisco Aironet 340 PCI 802.11b
KLE [Senel et al., 2007]	RSSI*	Transmissor	bidirecional	Sensores usando chip CC2420
LQE-Node [Gomes et al., 2017b]	RSSI	Nós dedicados para estimar	bidirecional (utilizando 2 nós dedicados)	Microchip MRF24J40MA ZigBee
REP [Rojas & Decotignie, 2018]	RSSI e LQI	Receptor	downlink	TelosB, com parâmetros definidos por simulação em Cooja
QIN [Qin et al., 2018]	RSSI	Desconhecido	downlink	Duas placas SDR NI USRP-2922
Gomes [Gomes et al., 2019]	RSSI e uma variação do PRR	Transmissor	downlink	Avalia a versão simplificada somente com RSSI. Usa dois microcontroladores PIC18LF4620 e dois transceivers MRF24J40MA
Four-bit [Fonseca et al., 2007]	PRR e RNP	Híbrido	bidirecional	TelosB e MicaZ motes
Triangle Metric [Boano et al., 2010]	PRR, SNR e LQI	Transmissor	uplink	Sensores usando chip CC2420
PRR ² [da Silva et al., 2017]	RSSI e dois níveis dePRR	Transmissor	uplink	Três placas SDR NI USRP B200
E2E-DDR [Sun et al., 2017b]	RSSI, PRR e ruído de fundo	Transmissor	uplink	Microchip baseado em TI-CC2530
CLQE [Ferng & Abdullah, 2019]	SNR, PRR e velocidade média do veículo	Desconhecido	Desconhecido	Simulação no NS-3
Demetri [Demetri et al., 2019]	ESP**, PRR e imagens geográficas	Desconhecido	Desconhecido	Placa Dragino LoRa shield v1.3 com chip RF96. Expected Signal Power (ESP), derivado do RSSI e SNR
Bio-inspired [Caleffi & Paura, 2009]	ETX	Híbrido	bidirecional	Simulação no NS-2
F-LQE [Baccour et al., 2010]	PRR e SNR	Transmissor	bidirecional	TelosB motes
4C (Foresee) [Liu & Cerpa, 2014]	RSSI, PRR e ETX	Transmissor	bidirecional	Motelab testbed (21 TMotes)
Anfis LQE [Vallejo et al., 2014]	RSSI, acelerômetro e parâmetros corporais	Receptor	downlink	Plataforma de sensores Shimmer utilizando microcontroladores TI MSP430, com parâmetros definidos no Matlab
LQL [Di Caro et al., 2013]	PRR e LWPR	Híbrido	bidirecional	Simulação no NS-3 e parcialmente no testbed Indriya
WNN-LQE [Sun et al., 2017a]	SNR e PRR	Desconhecido	Desconhecido	Coleta de dados com microchip TI CC2530 e simulação em Matlab
Huang [Huang et al., 2019]	LQI, SNR e PRR	Híbrido	bidirecional	Simulações com OMNeT++ e a extensão MiXiM
Sindjounng [Sindjounng & Minet, 2019]	RSSI e PDR	Receptor	downlink	Testbed Grenoble com nós M3
RVFL [Xue et al., 2020]	SNR	Híbrido	bidirecional	Microchips baseados no microchip TI-CC2530

3.4 Conclusão

Os trabalhos apresentados nesta seção, evidenciam a relevância da temática no contexto de redes de computadores e transmissão de dados. Apesar dos diversos estimadores propostos na literatura, observa-se que ainda há lacunas quando se analisa os estimadores com processos de aprendizagem, o que pode fornecer uma estimativa mais acurada. Ademais, nenhum dos estimadores de qualidade do enlace previamente descritos consideram *concept drift* em suas estratégias de decisão.

Este trabalho se destaca dos desenvolvidos anteriormente por adotar informações de múltiplas camadas de rede aliado a um sistema de aprendizado de máquina que utiliza o tratamento de *concept drift* para fornecer uma estimativa mais precisa em diversos cenários. Diferentemente dos métodos mencionados previamente nesta seção, a proposta neste trabalho também considera simultaneamente RSSI, SNR, PRR, latência e a taxa de pacotes transmitidos por pacotes recebidos (TPRP). As avaliações dos estimadores propostos são conduzidas em experimentos reais usando um *testbed* de USRPs (modelo Ettus B200), o que conduz a experimentos mais precisos, fidedignos, diretos e replicáveis do que simulações. O método proposto não requer alterações no lado receptor, permitindo sua adoção em vários sistemas de comunicação M2M.

Capítulo 4

Estimador PRR²

Este capítulo apresenta um novo estimador de qualidade de enlace para comunicação máquina a máquina, denominado PRR². O estimador PRR² emprega duas métricas, RSSI (*Received Signal Strength Indication*) e PRR (*Packet Reception Ratio*), bem como dois níveis de PRR. O uso de duas janelas deslizantes de PRR captura variações de qualidade de enlace no curto prazo também considerando o longo prazo. O capítulo está dividido em duas seções: na primeira é descrito o estimador proposto e na segunda é demonstrado o seu funcionamento.

4.1 Descrição do PRR²

O PRR² é um estimador de qualidade de enlace originalmente projetado para executar em dispositivos que possuem baixo poder de processamento ou que precisem limitar a computação em razão de limitações de energia. O estimador PRR² é baseado em métricas provenientes de duas camadas de rede, sendo uma das métricas o PRR tradicional, porém com uma leitura em dois níveis, de curto e longo alcances, razão da nomenclatura do estimador. O PRR² está configurado para ser acionado sempre que ocorre uma demanda por envio de mensagens.

A primeira métrica que o estimador considera é o RSSI do lado transmissor. Especificada uma determinada frequência, esta métrica fornece um indicativo do nível de potência na cobertura da antena do transmissor. O RSSI é utilizado no algoritmo para identificar situações em que os enlaces estão momentaneamente concorridos e para preencher a lacuna quando o PRR não pode ser calculado de forma confiável, o que ocorre quando não há pacotes suficientes para estimar o PRR, ou quando os intervalos entre os novos pacotes são muito longos. Durante esse primeiro

modo de operação, somente com a métrica RSSI, dizemos que o PRR² está em seu 1º estágio de operação.

A segunda métrica do PRR² é o PRR com leitura em dois níveis. Um nível é calculado a partir de uma janela móvel pequena, enquanto outro é computado sobre uma janela móvel longa, conforme exibido na Figura 4.1. A utilização de dois níveis de PRR permite capturar variações em curto período de tempo, enquanto mantém uma informação de longo prazo. Isso é útil em muitas situações: (i) para evitar estimativas erradas devido a uma oscilação rápida do ruído de fundo; (ii) para reduzir os efeitos do padrão de radiação anisotrópico das antenas [Zhou et al., 2006], uma vez que as irregularidades da rádio-propagação podem ser mitigadas usando dois níveis de confirmação, a curto e longo prazos. A definição do tamanho das janelas é passada por meio de um parâmetro durante a execução do algoritmo estimador. Neste modo de funcionamento, com RSSI e PRRs de curto e longo alcances, dizemos que o PRR² está em seu 2º estágio de operação.

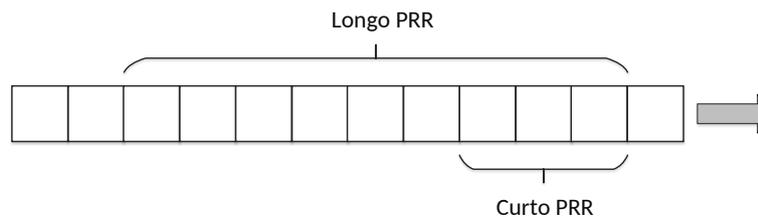


Figura 4.1. Esquema de PRR²: janelas PRR curta e longa

4.1.1 Parâmetros de entrada e saída

O estimador PRR² possui como principais entradas (i) o valor do RSSI do lado transmissor, (ii) o pacote *ack* (*acknowledgment*) proveniente do equipamento receptor, e (iii) uma notificação de tempo extrapolado de espera pelo *ack*.

A fim de estabelecer uma métrica de fácil associação com qualidade do enlace, foi assumido que quanto mais próximo de *zero* uma estimativa for, pior está a qualidade do enlace, ao passo que quanto mais próxima de 1 for a estimativa, melhor a qualidade do enlace. Para isso, o valor de saída do estimador é normalizado entre o intervalo fechado de 0 e 1. A Figura 4.2 ilustra as entradas e saída do estimador PRR². A operação é modular e segue os seguintes passos: suavização, cálculo do PRR, normalização e computação, descritos a seguir.

Suavização: considerando que o meio sem fio apresenta alta variabilidade da qualidade do enlace, para suavizar flutuações de curto prazo, tanto o PRR quanto

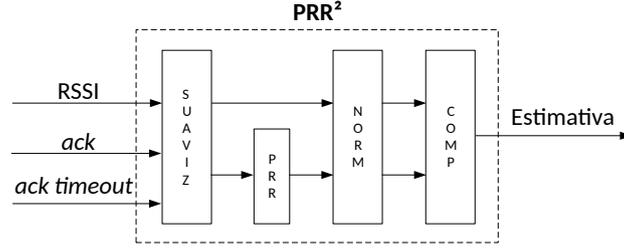


Figura 4.2. Entradas e saída do estimador proposto PRR²

o RSSI são computados utilizando média móvel exponencial (EMA - *Exponential Moving Average*), exceto quando não há pacotes suficientes para computar de forma confiável a EMA. Nesse caso, a média móvel simples (SMA - *Simple Moving Average*) é utilizada, conforme expresso a seguir:

$$smooth(W) = \begin{cases} \frac{\sum_{i=1}^{|W|} w_i}{|W|}, & \text{if } |W| \leq k \\ \sum_{i=1}^{|W|} (1 - \alpha) \alpha^{i-1} w_i, & \text{if } |W| > k \end{cases} \quad (4.1)$$

onde W é o vetor da janela de observação, $|W|$ representa o tamanho da janela, w_1 é a medição mais recente, $w_{|W|}$ é a menos recente, k representa o limiar para mudança de EMA e SMA, e α é o fator de suavização. Em razão do tamanho da janela de observação ser dinâmico, caso o valor de α fosse calculado baseado em um conjunto de dados pretéritos e com uma janela fixa, o resultado não seria o mais adequado para uso em tempo real. Por isso foi adotada a estratégia de definir o valor de α dinamicamente, recalculado a cada nova transmissão de dados, e é computado como $\alpha = 2/(|W| + 1)$, de acordo com Nahmias & Olsen [2015]. Esta abordagem de suavização exponencial minimiza flutuações aleatórias recentes, provendo uma estimativa mais estável.

Cálculo do PRR: nosso algoritmo calcula dois PRRs a partir do pacote *ack* recebido e da indicação de tempo de espera do *ack* extrapolado. Um é o PRR computado sobre uma pequena janela de dados, e outro é o computado sobre uma longa janela de dados. O PRR é obtido utilizando a Equação 4.2, onde i e j representam, respectivamente, os índices de início e de fim da janela de observação.

$$PRR_{i,j} = \frac{ack_{(i,j)}}{ack_{(i,j)} + timeout\ flag_{(i,j)}} \quad (4.2)$$

Normalização: por convenção, os valores estimados por PRR² são normalizados entre $[0, 1]$, sendo que valores mais altos indicam melhor qualidade do enlace. O processo de normalização dos valores de PRR é direto, visto que o PRR já fornece uma estimativa nesse mesmo intervalo. A normalização do RSSI requer considerações adicionais: o intervalo pode representar o maior e menor valor de RSSI em uma implantação física. Além disso, eles devem ser calibrados baseados na capacidade de cada equipamento *transceiver*. Mais detalhes do processo de normalização para um *transceiver* específico são descritos na Seção 6.2.2.

Computação: a escolha por uma métrica para usar o PRR² é baseada na equação 4.3. *longoPRR* representa o PRR computado sobre a janela móvel longa, e *curtoPRR* representa o PRR calculado usando a janela móvel curta. *estRSSI* é o valor de RSSI observado no lado do transmissor, normalizado para um valor positivo entre o intervalo fechado 0 e 1. Neste estágio, todos os três valores já estão normalizados entre 0 e 1:

$$PRR^2 = \min(\textit{longoPRR}, \textit{curtoPRR}, \textit{estRSSI}) \quad (4.3)$$

Visando propiciar uma transmissão de dados nas piores condições do canal de comunicação, o estimador considera a pior das três estimativas observadas, e por isso a saída é selecionada a partir do menor dos valores de *longoPRR*, *curtoPRR* e RSSI estimado. Dessa forma, pode-se compreender que a Equação 4.3 atua de forma conservadora.

4.2 Funcionamento do PRR²

A Figura 4.3 apresenta o mecanismo geral de operação do estimador PRR²: inicialmente (1), antes da transmissão da primeira mensagem, o módulo extrai as informações instantâneas do RSSI e as suaviza utilizando média móvel simples - SMA (2). Essa será a primeira estimativa da qualidade do enlace, visto que ainda não foram transmitidos ou recebidos pacotes, impossibilitando computar um valor de PRR. Em seguida, é calculada a estimativa, valor que pode ser utilizado por módulos como controle de potência ou de controle de taxa de transmissão (3 e 4).

Após a primeira transmissão de pacote, o módulo PRR² aguarda uma mensagem de confirmação (*ack*) ou um aviso de perda de pacote para iniciar o cômputo

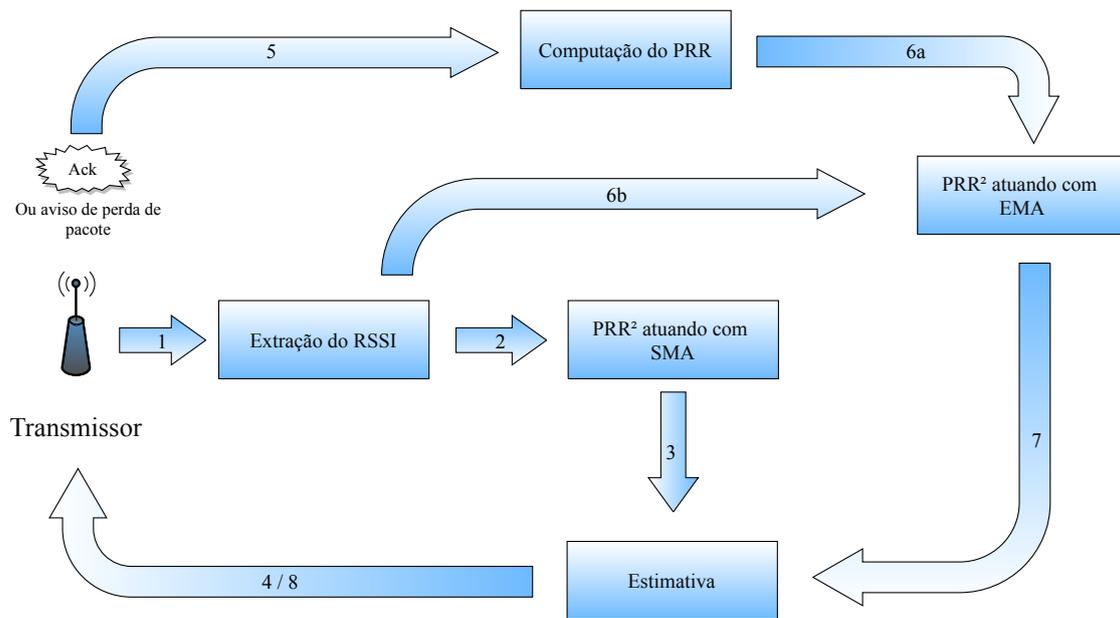


Figura 4.3. Etapas de operação do PRR²

do PRR. Contudo, enquanto um *ack* não for recebido e não houver sinalização de perda do pacote transmitido, novas tentativas de transmissão podem ser realizadas, utilizando uma estimativa de qualidade do enlace atualizada proveniente da métrica RSSI.

Após o recebimento do primeiro *ack* ou aviso de perda de pacote (5), dá-se início à formação da janela de PRR (6a). De posse do mais recente valor médio do RSSI (6b), o algoritmo PRR² suaviza todos os valores com média móvel exponencial (EMA). Nesse instante o algoritmo começa a operar em sua funcionalidade plena e a estimativa é calculada (7) conforme a Equação 4.3. Essa última operação garante que ruídos externos representativos sejam imediatamente considerados na estimativa de qualidade do enlace. Em seguida, a estimativa é enviada ao transmissor (8) para realizar de fato a transmissão da nova mensagem.

Enquanto não houver o reinício do dispositivo, o algoritmo estará sempre em seu funcionamento pleno (sem passar pelas etapas 2 e 3 da Figura 4.3).

O algoritmo do PRR² foi desenvolvido de forma a calcular, a partir de um conjunto de variáveis, a estimativa de qualidade do enlace sempre que houver uma demanda de envio de pacote. Essas variáveis podem ser atualizadas por eventos de recebimento de *ack* ou de sinalização de perda de pacote (*timeout*), independente de ocorrer, ou não, uma demanda de transmissão. Essa estratégia garante que mesmo em grandes intervalos de transmissão a estimativa seja calculada a partir de

dados atualizados. As entradas e saída do PRR² estão demonstradas na Figura 4.2, apresentando os principais módulos responsáveis pelo funcionamento do estimador.

O funcionamento geral da principal função (*Computação*) do estimador PRR² pode ser visto no Algoritmo 1 a seguir. Os atributos de entrada são os parâmetros *tipo*, *maxTamSerie* e *estRSSI*. O primeiro atributo de entrada identifica o *tipo* de acionamento: se por meio de um recebimento de *ack*, ou por meio da sinalização de tempo extrapolado (*timeout*). O segundo atributo de entrada é o parâmetro *maxTamSerie* que representa o tamanho máximo da série (janela de observação), e o terceiro atributo, *estRSSI*, representa o valor estimado do RSSI.

Algoritmo 1: PRR² - Computação

Entrada: *tipo, maxTamSerie, estRSSI*
Saída: estimativa

```

1 início
2   se (envio ou ack ou timeout) então
3     |   estimativa ← min(calcPRR(tipo,maxTamSerie), estRSSI)
4     fim
5 fim
6 retorna estimativa

```

Uma importante função no PRR² é a *calcPRR*, responsável por montar as séries de *ack* recebidos ou *timeouts* e calcular o *curtoPRR* e o *longoPRR*. O atributo *tipo* identifica o *tipo* de acionamento. Caso um *ack* seja recebido, o algoritmo adiciona o valor 1 às séries relacionadas ao *ack*; caso a sinalização de *timeout* seja recebida, é adicionado o valor *zero* às séries *ack*. *maxTamSerie* é utilizado em várias funções do PRR² diretamente, porém, nesta função *calcPRR* seu valor é multiplicado por *dois* nas séries que proverão o cálculo do *longoPRR*. Essa estratégia faz com que o tamanho da série representada por *longoPRR* seja duas vezes maior que aquela representada por *curtoPRR*. Essa função está contida no Algoritmo 2.

Os algoritmos completos do PRR², desenvolvidos na linguagem Python, podem ser verificados em repositório *online*¹.

O funcionamento do PRR² sob a ótica das operações realizadas durante as trocas de mensagens pode ser visualizado no diagrama de sequências da Figura 4.4. Para ilustrar o funcionamento completo do sistema de comunicação, foi incorporado ao diagrama um módulo de controle de potência, responsável por receber a estimativa calculada e ajustar a potência utilizada na transmissão. Os experimentos

¹Endereço do repositório GIT: <https://bitbucket.org/wendley/gr-lqe>

Algoritmo 2: calcPRR

Entrada: *tipo, maxTamSerie***Saída:** *curtoPRR, longoPRR*

```

1 início
2   se tipo = ack recebido então
3     serieACK ← addSerie(serieACK, 1, maxTamSerie)
4     seriePCK ← addSerie(seriePCK, 1, maxTamSerie)
5     serieLACK ← addSerie(serieLACK, 1, maxTamSerie * 2)
6     serieLPCK ← addSerie(serieLPCK, 1, maxTamSerie * 2)
7   senão se tipo = timeout então
8     serieACK ← addSerie(serieACK, 0, maxTamSerie)
9     seriePCK ← addSerie(seriePCK, 1, maxTamSerie)
10    serieLACK ← addSerie(serieLACK, 0, maxTamSerie * 2)
11    serieLPCK ← addSerie(serieLPCK, 1, maxTamSerie * 2)
12  fim
13  emaSA ← ema(serieACK, tam(serieACK))
14  emaSP ← ema(seriePCK, tam(seriePCK))
15  emaSLA ← ema(serieLACK, tam(serieLACK))
16  emaSLP ← ema(serieLPCK, tam(serieLPCK))
17  curtoPRR ← emaSA div emaSP
18  longoPRR ← emaSLA div emaSLP
19 fim
20 retorna curtoPRR, longoPRR

```

práticos com os *hardwares* programáveis também executam o controle de potência com essa característica.

No lado transmissor estão os *atores* denominados Emissor, PRR² 1º e 2º estágios. No receptor está o *ator* homônimo. Este é responsável por receber as mensagens, verificar o destinatário e o CRC (*Cyclic redundancy check*), e responder ao transmissor com um pacote *ack*.

4.3 Conclusão

Este capítulo apresentou o PRR², um estimador de qualidade de enlace de múltiplas camadas que adota a estratégia de duas janelas de PRR para fornecer uma estimativa mais estável e precisa. O estimador PRR² combina RSSI, PRR de curto alcance e PRR de longo alcance. Observa-se que o equipamento receptor não precisa de modificações ou adaptações em algoritmos para funcionar com o PRR². Foi utilizada

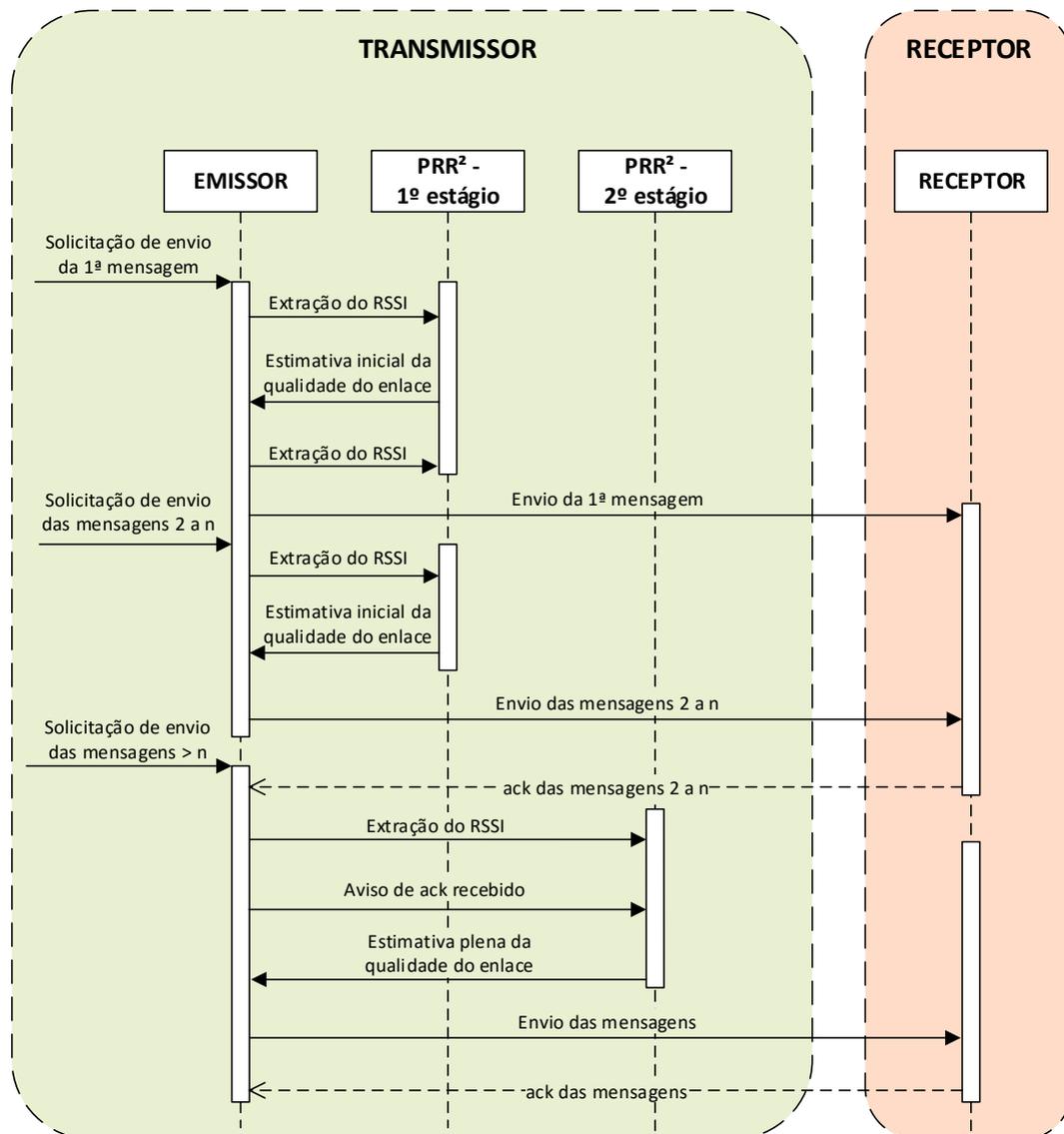


Figura 4.4. Diagrama de sequência do PRR²

uma plataforma de teste com experimentos reais para o processo de desenvolvimento. Os resultados dos experimentos com o estimador PRR² estão apresentados no Capítulo 6.

Capítulo 5

Estimador LQM

Este capítulo apresenta um novo estimador de qualidade de enlace para comunicação máquina a máquina baseado em aprendizagem de máquina. O capítulo está dividido em três seções: na primeira é apresentado o estimador desenvolvido, denominado LQM (*Link Quality Estimator using Machine Learning with Concept Drift adaptation*); na segunda, é descrita a metodologia para escolha dos melhores parâmetros de entrada, considerando a correlação entre eles. Por fim, na terceira seção, é apresentado como o algoritmo de aprendizagem de máquina a ser utilizado foi selecionado e como ocorre o processo de identificação de *concept drifts*.

Para identificarmos as métricas mais relevantes para o processo de estimação com aprendizagem de máquina, inicialmente foi realizada uma seleção dos melhores algoritmos e melhores entradas, conforme metodologia explicada na próxima seção.

5.1 LQM - estimador

Esta seção apresenta uma nova abordagem para um estimador de qualidade de enlace projetado para comunicação máquina-a-máquina, o *Link Quality Estimator using Machine Learning with Concept Drift adaptation* (LQM). O estimador LQM é multicamadas e produz estimativas a partir de RSSI, SNR, PRR, latência e TPRP (*Transmitted packets per received packets*), o que proporciona maior robustez. As várias camadas adicionam informações de qualidade do enlace que não podem ser estimadas isoladamente. Por exemplo, um estimador que considere apenas o RSSI local não consegue capturar a qualidade de todo o enlace, enquanto um estimador usando apenas o PRR pode não ser capaz de capturar as flutuações momentâneas da qualidade do enlace nas proximidades do transmissor.

O estimador LQM é executado no lado do emissor e, de maneira semelhante ao estimador PRR², é configurado para ser acionado sempre que ocorrer uma demanda de envio de mensagens. Em seguida, os dados coletados e acumulados dentro da janela de observação são submetidos ao algoritmo de detecção de desvio ADWIN (*Adaptive Windowing*). Conforme descrito na Seção 2, esse algoritmo utiliza janelas deslizantes de tamanhos variáveis que são recalculadas de acordo com o grau observado dos desvios nos dados contidos nas janelas [Bifet & Gavalda, 2007].

Na figura 5.1 é ilustrado um esquema simplificado do mecanismo de operação do LQM. A operação envolve diretamente dois módulos: o módulo de detecção de Concept Drift (CD) e o módulo de *Machine Learning* (ML). Os dados de entrada são processados primeiro pelo algoritmo ADWIN para identificar CD e também entregues, na sequência, ao algoritmo de ML. Se houver detecção de CD, o algoritmo de ML será retreinado com uma nova base. A saída do algoritmo de ML é a estimativa da qualidade do enlace.

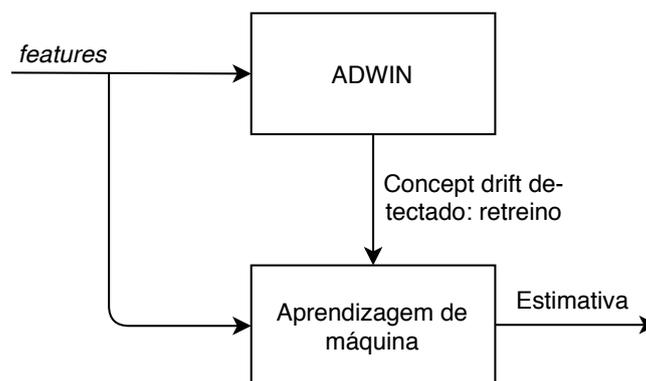


Figura 5.1. Esquema de operação do LQM

Os passos de operação do emissor com o LQM podem ser observados na Figura 5.2. Inicialmente, o sistema carrega o arquivo de treinamento (*joblib*). Quando existe uma demanda para enviar um novo pacote, a ocorrência de *concept drift* (CD) é verificada de acordo com as entradas do algoritmo. Se um CD for identificado, a máquina será retreinada utilizando os dados atualizados e projetará a estimativa. Se nenhum CD for identificado, a máquina estima a saída diretamente.

Para o processo de retreino do LQM, um pequeno banco de dados é construído e armazenado dinamicamente em tempo de execução, e será utilizado apenas se o LQM com estimador de CD detectar *concept drift*. O tamanho do banco de dados é variável e será especificado na seção de avaliação do LQM. O LQM treina novamente o modelo com o banco de dados disponível no momento da detecção do

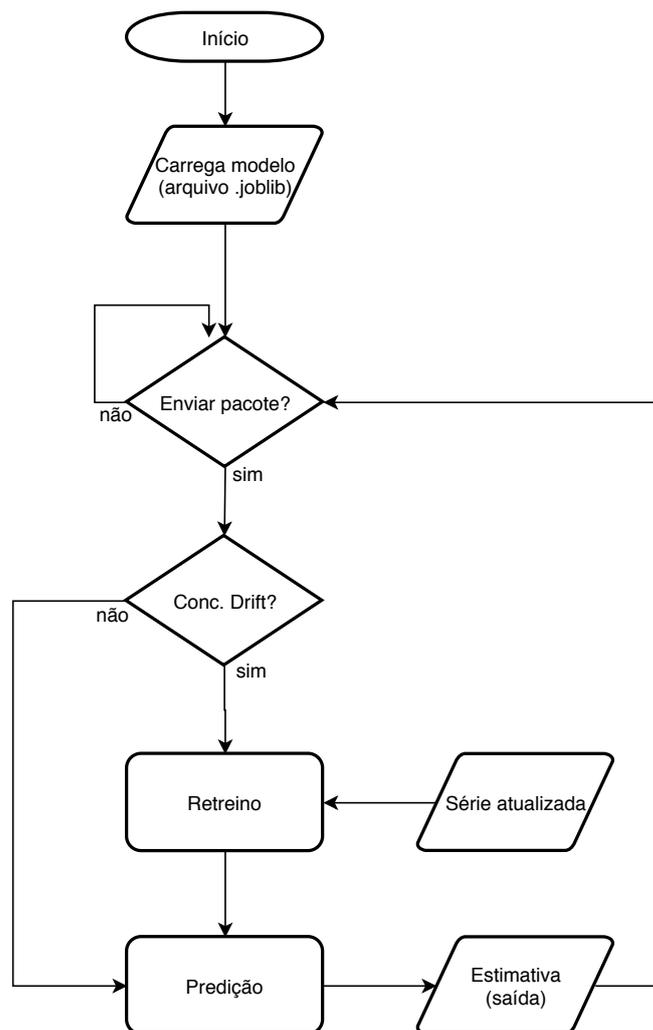


Figura 5.2. Fluxo de funcionamento do emissor com LQM

CD (incluindo a amostra que indicou o CD). Para a avaliação, foram consideradas as seguintes métricas:

Taxa de entrega de pacotes: é um indicador da qualidade da comunicação percebida pelo usuário. A taxa de entrega de pacotes pode ser definida como a proporção entre o número de pacotes recebidos com sucesso e o número de pacotes enviados.

Índice do ganho de transmissão: é a quantidade de energia gasta por transmissão (um ganho mais alto exige uma maior potência de transmissão, implicando em maior gasto de energia).

Relação de pacotes transmitidos por pacotes recebidos (TPRP): representa a proporção entre os pacotes enviados, incluindo as retransmissões, e os recebidos. Pode ser compreendido como uma indicação de quão eficiente foi o pro-

cesso de entrega, já que para um pacote recebido, em algumas situações, mais de uma transmissão do mesmo pacote pode ser necessária.

5.2 Obtenção das entradas

As entradas para a estimação de qualidade são coletadas a partir de três informações: RSSI, SNR e pacotes ACK recebidos. Os dois primeiros são obtidos localmente em cada nó de transmissão, enquanto o pacote ACK é o pacote de confirmação enviado pelo nó de recebimento. O PRR e o TPRP são atualizados à medida que pacotes ACK são recebidos.

Uma importante métrica que o LQM considera como entrada é o RSSI (*Received Signal Strength Indicator*) coletado no lado transmissor. Amostrada na mesma frequência de transmissão, essa métrica fornece uma indicação do nível de potência detectado pela antena do receptor, no lado transmissor, estratégia possível em razão do USRP possuir essas antenas independentes. A métrica RSSI é utilizada no LQM para auxiliar na identificação de *links* instáveis, quando o PRR não puder ser calculado com precisão (ou seja, quando não houver pacotes suficientes para estimar o PRR ou quando o intervalo entre novos pacotes for muito longo).

A faixa de valores de RSSI varia de acordo com o modelo USRP. De acordo com a referência do manual do dispositivo analógico AD9361 ¹, embutido no USRP B210, a sensibilidade mínima teórica do RSSI é -128 dB, enquanto o valor máximo teórico do RSSI é 0 dB. No entanto, esses valores dificilmente são alcançados na prática. Para identificar um valor apropriado para o experimento, a faixa RSSI foi calibrada para fornecer uma taxa de entrega equivalente ao PRR em condição sem ruído artificial. O RSSI mínimo prático observado foi de -128 dB, enquanto o máximo foi de -68 dB.

Como o meio sem fio varia continuamente, para suavizar as flutuações de curto prazo, tanto o PRR quanto o RSSI são calculados a partir de médias móveis, nesse caso, a Média Móvel Exponencial (EMA). A EMA é adotada por ser computacionalmente simples, eficiente em termos de memória e amplamente utilizada na literatura Woo & Culler [2003]. O cálculo da EMA ocorre sempre que houver demanda por uma transmissão de pacotes. Vale ressaltar que, nas primeiras transmissões de pacotes, quando não há pacotes suficientes para calcular com segurança a EMA, é utilizada a Média Móvel Simples (SMA). Isso porque o algoritmo EMA requer um número mínimo de valores para ser resolvido corretamente. A equação 4.1

¹Disponível em http://www.analog.com/media/en/technical-documentation/user-guides/AD9361_Reference_Manual_UG-570.pdf

utilizada pelo estimador PRR² na Seção 4 é a mesma utilizada pelo estimador LQM e mostra a fórmula matemática para calcular a suavização EMA de um conjunto de observações ($smooth(W)$).

A métrica TPRP é útil para medir a dificuldade de transmitir pacotes com sucesso na camada de enlace. Seu valor é sempre maior que 1 e indica se são necessárias retransmissões frequentes para entregar um pacote. Por exemplo, um link sem perdas geraria um TPRP de 1. Por outro lado, se todos os pacotes forem perdidos e o remetente tiver que transmitir dados três vezes a cada tentativa de envio, o TPRP terá o valor 3.

Todas as entradas são normalizadas para melhorar a precisão do modelo de aprendizado de máquina. O processo de normalização dos valores de PRR é direto, uma vez que o PRR já se encontra dentro do intervalo $[0, 1]$. A normalização dos valores de RSSI exige uma análise mais aprofundada: o intervalo deve representar os valores mais altos e mais baixos de RSSI em uma implantação real; para isso, dezenas de amostras e testes foram mapeados com ausência e presença de ruído artificial para estimar os valores mais apropriados. Além disso, eles devem ser calibrados com base nas capacidades de cada transceptor, o que é dependente do modelo do *hardware* em uso e, para isso, calibrações empíricas foram realizadas com base nas especificações do equipamento. Em nosso experimento, a calibração está de acordo com a referência do chip AD9361, conforme citada anteriormente.

5.3 Definição dos algoritmos de aprendizagem de máquina

Esta seção descreve a seleção das características (*features*) para o modelo ML, bem como a avaliação para identificar o melhor algoritmo de ML para usar no LQM. Os dados foram coletados usando o *testbed* Futebol² com dispositivos USRP. O protocolo ZigBee (IEEE-802.15.4) foi utilizado para transmissão de dados e os nós foram posicionados como mostrado na Figura 6.13. As transmissões foram realizadas sem a presença de ruído artificial e também com a injeção de ruído artificial usando o comando GNU Radio `uhd_siggen -- gaussian`.

Os nós utilizados para coletar os dados para o treinamento foram aqueles identificados como TX1, TX2, TX3 e TX4 na Figura 6.13. Os dados coletados estão disponíveis para download³. Foram transmitidos pacotes de 100 bytes em

²<http://www.futebol.dcc.ufmg.br/>

³<https://github.com/wendley/sdr-data>

intervalos de 250 ms com e sem a adição de ruído artificial, com combinações de transmissores individuais e múltiplos, totalizando 14.540 amostras. Esses dados são usados apenas uma vez para o treinamento inicial; após esta etapa, o modelo de aprendizado é salvo em um arquivo no formato *.joblib*. Os dados de entrada para os algoritmos de aprendizado de máquina são aqueles obtidos pelo equipamento de transmissão sem fio USRP, sendo eles *rsssi*, *prrr*, *prrr2*, *snr*, *taxadeEntrega*, *TPRP*, *latência* e *potência*.

5.3.1 Analisando a série temporal

Inicialmente, como estamos lidando com variações de sinais ao longo do tempo, verificamos se precisaríamos empregar técnicas de aprendizado de máquina específicas para séries temporais [Qian & Chen, 2019]. Esta seção detalha nossa análise. Primeiro, *plotamos* a Estimativa de densidade do Kernel (KDE) com o ajuste gaussiano de cada entrada. O KDE é um método não paramétrico de estimar a função de densidade de probabilidade de uma variável aleatória, tornando-se útil para visualizar a forma das distribuições de dados. Um *rugplot* é exibido como marcas ao longo do eixo x para exibir os dados de uma única variável quantitativa (isto é, desenha um pequeno tique vertical em cada observação de dados).

A Figura 5.3 mostra a distribuição de dados para *RSSI* e *SNR*. O KDE do *RSSI* e do *SNR* tem uma distribuição bimodal, que geralmente indica dois grupos diferentes de população de dados: o mais à esquerda, indicando ausência de transmissão na frequência analisada e ao alcance do equipamento, enquanto o pico mais à direita indica uma possível transmissão. Esse padrão sugere persistência na série, indicando que pode ser uma série não estacionária, que pode ser confirmada ou rejeitada pelo teste da Função de Correlação Automática (ACF) e Dickey-Fuller Aumentado (ADF - *Augmented Dickey-Fuller*), discutido adiante.

O ACF ou Correlograma resume a correlação de uma série temporal por vários *lags* [Hyndman & Athanasopoulos, 2018]. Quando todos os dados coletados são considerados, o gráfico do ACF de *rsssi*, *snr*, *prrr*, *taxadeEntrega*, *TPRP* e *latencia* indica forte persistência em todos os atrasos (*lags*) (o lento decaimento da correlação ao longo dos *lags*, após o *lag* 0), como mostra a Figura 5.4. Finalmente, sabendo-se que algoritmos de aprendizagem de máquina desenvolvidos para séries temporais possuem maior eficiência nas séries que originalmente apresentam estacionariedade [Qian & Chen, 2019], procedemos a análise para verificar se a série temporal trabalhada é estacionária. Para isso, foi utilizado o teste ADF. A Tabela 5.1 mostra os resultados do teste ADF com o indicativo na última linha se

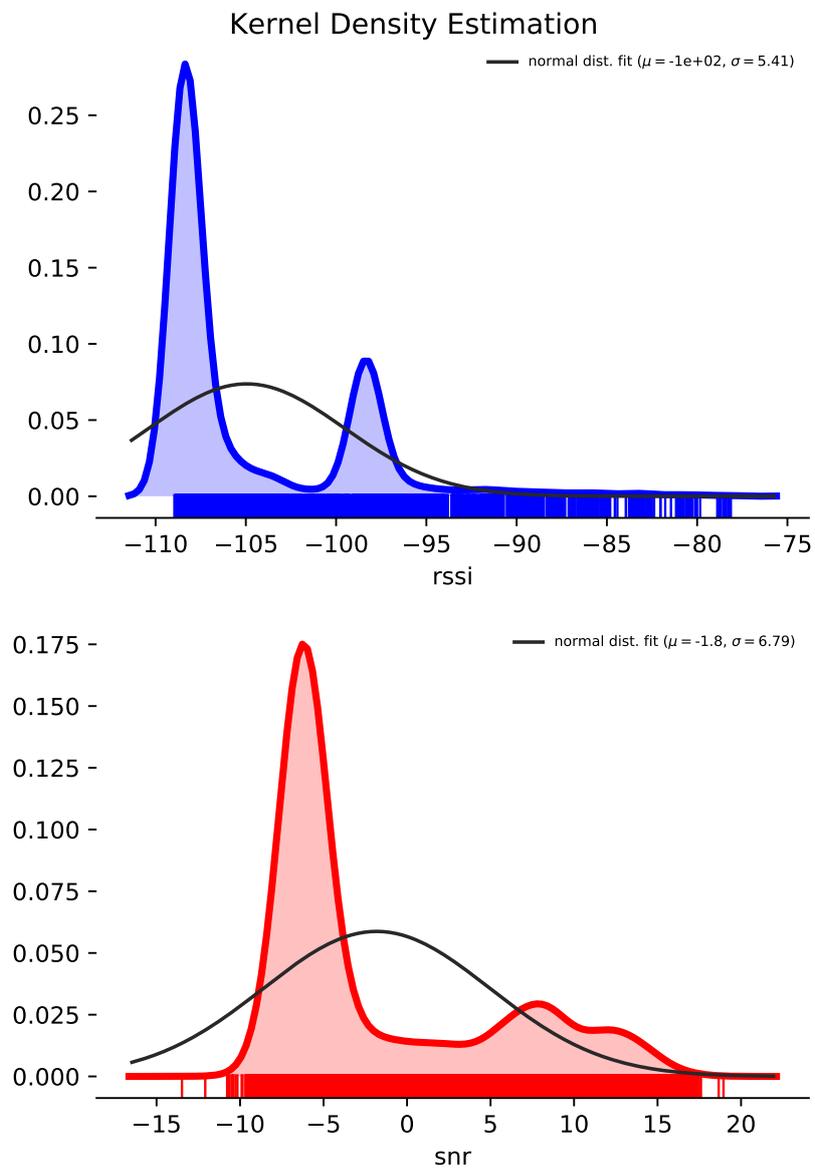


Figura 5.3. KDE com linha da forma gaussiana: KDE do RSSI na imagem superior e KDE do SNR na imagem inferior

cada série se comporta como uma série estacionária. O processo é estacionário apenas se todas as entradas forem estacionárias [Qian & Chen, 2019]. Como visto na tabela, o RSSI não é estacionário e, portanto, a série temporal não é estacionária. Esse resultado indica que os algoritmos de aprendizado de máquina personalizados para séries temporais podem não apresentar melhores resultados se aplicados para a série temporal analisada.

Tabela 5.1. Teste estatístico ADF

<i>Augmented Dickey-Fuller (ADF)</i>						
	RSSI	SNR	PRR	entrega	TPRP	latência
Estatísticas do Teste ADF	-1.8809	-9.5566	-4.6541	-2.4969	-6.7004	-4.9604
<i>P-Value</i>	0.3410	2.48e-16	0.0001	0.1162	3.90e-09	0.000027
# Lags	63	63	41	46	62	63
# Observações	71407	71407	71407	71407	71407	71407
Valor crítico (5%)	-2.8615	-2.8615	-2.8615	-2.8615	-2.8615	-2.8615
Valor crítico (1%)	-3.4304	-3.4304	-3.4304	-3.4304	-3.4304	-3.4304
Valor crítico (10%)	-2.5667	-2.5667	-2.5667	-2.5667	-2.5667	-2.5667
A série temporal é estacionária?	<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>

5.3.2 Selecionando as entradas do modelo

O primeiro passo do processo de avaliação do modelo de *machine learning* consiste em escolher as características (*features*) certas para o modelo. O processo de seleção realizado remove as características altamente correlacionadas, já que eles não contribuem com novas informações para o modelo, além de remover as características que reduzem a precisão do modelo.

A correlação entre as entradas ajuda a identificar aquelas que são mais representativas para a saída desejada (*target*). O cálculo da correlação implementado é o coeficiente de correlação de Pearson e de Spearman [Benesty et al., 2009]. Esses coeficientes assumem valores entre -1 e 1, onde um valor 1 significa uma correlação positiva perfeita entre os recursos e um valor -1 uma correlação negativa perfeita entre as entradas. O valor 0 significa que não há correlação entre as entradas. O resultado dessa análise é ilustrado no mapa de calor da Figura 5.5. Com o correlograma, pretendemos identificar quais recursos contribuem mais para causar efeitos nos *targets* taxadeEntrega e *TPRP*, ambos sinalizados nas figuras com retângulos

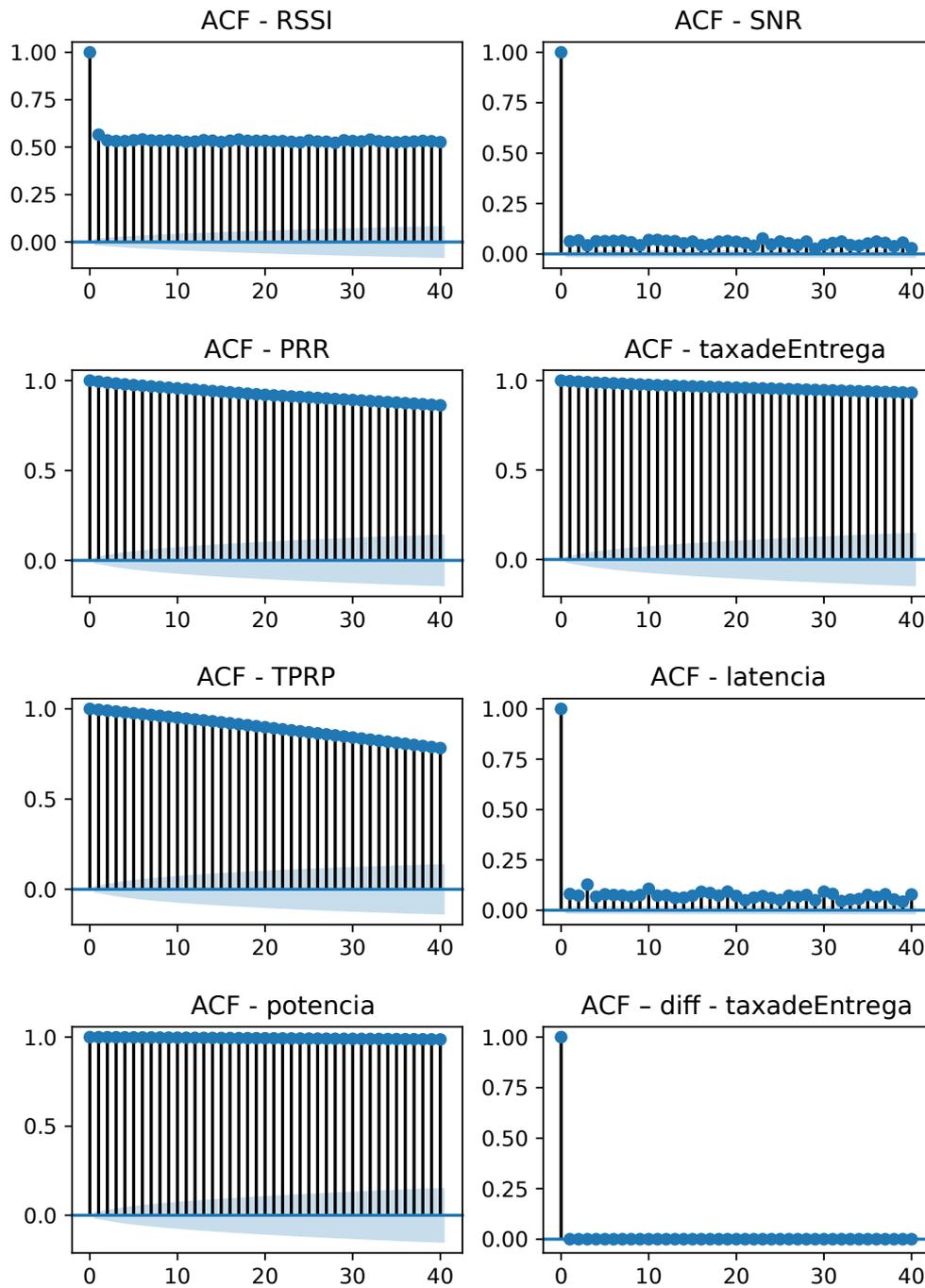


Figura 5.4. Função de autocorrelação

vermelhos. Em seguida, prosseguimos com a análise da matriz de correlação entre eles, como mostrado na Tabela 5.2.

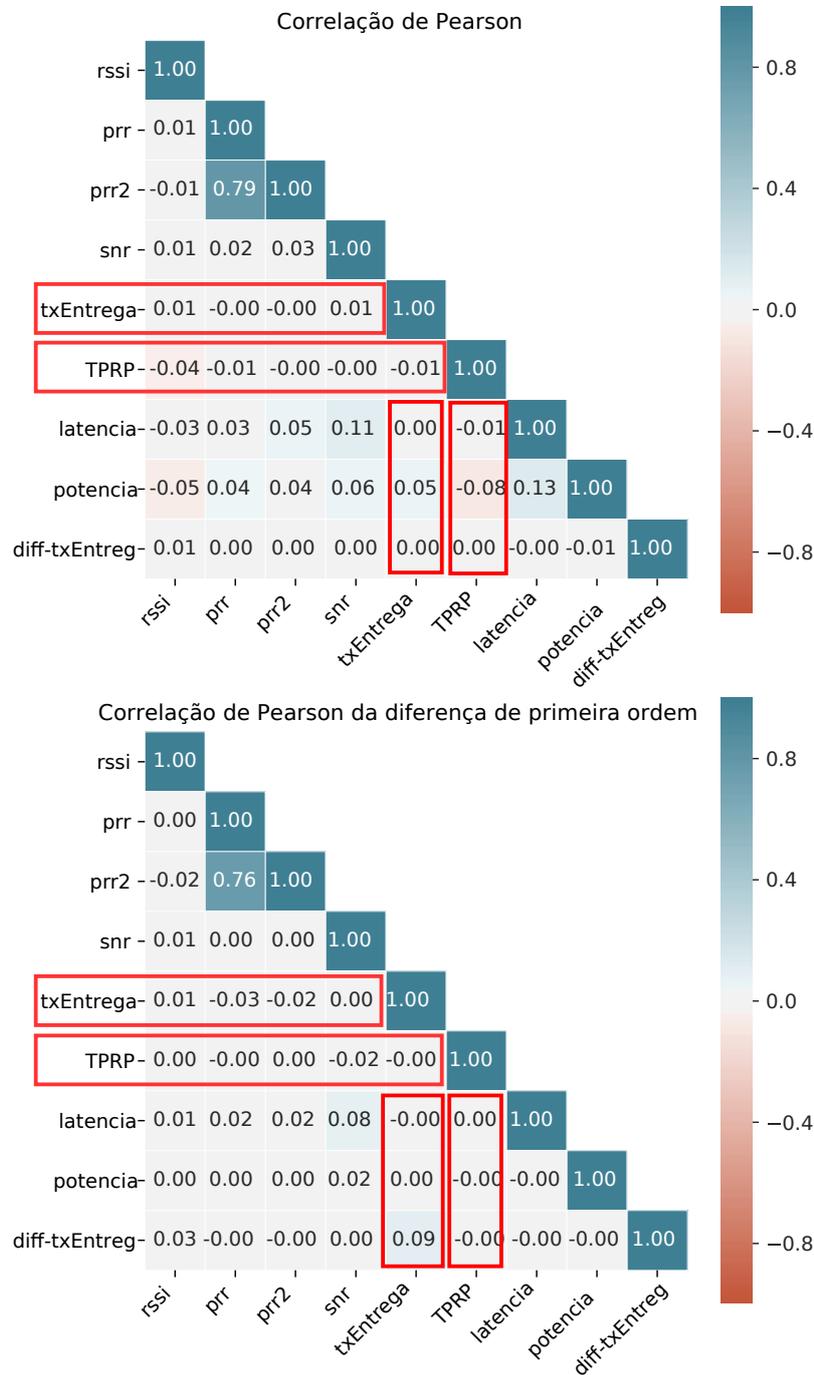


Figura 5.5. Correlação de Pearson para 8 features

A partir da análise dos mapas de calor das figuras 5.5 e 5.6, considerando que um valor de 0 implica que não há correlação linear entre os recursos, as correlações

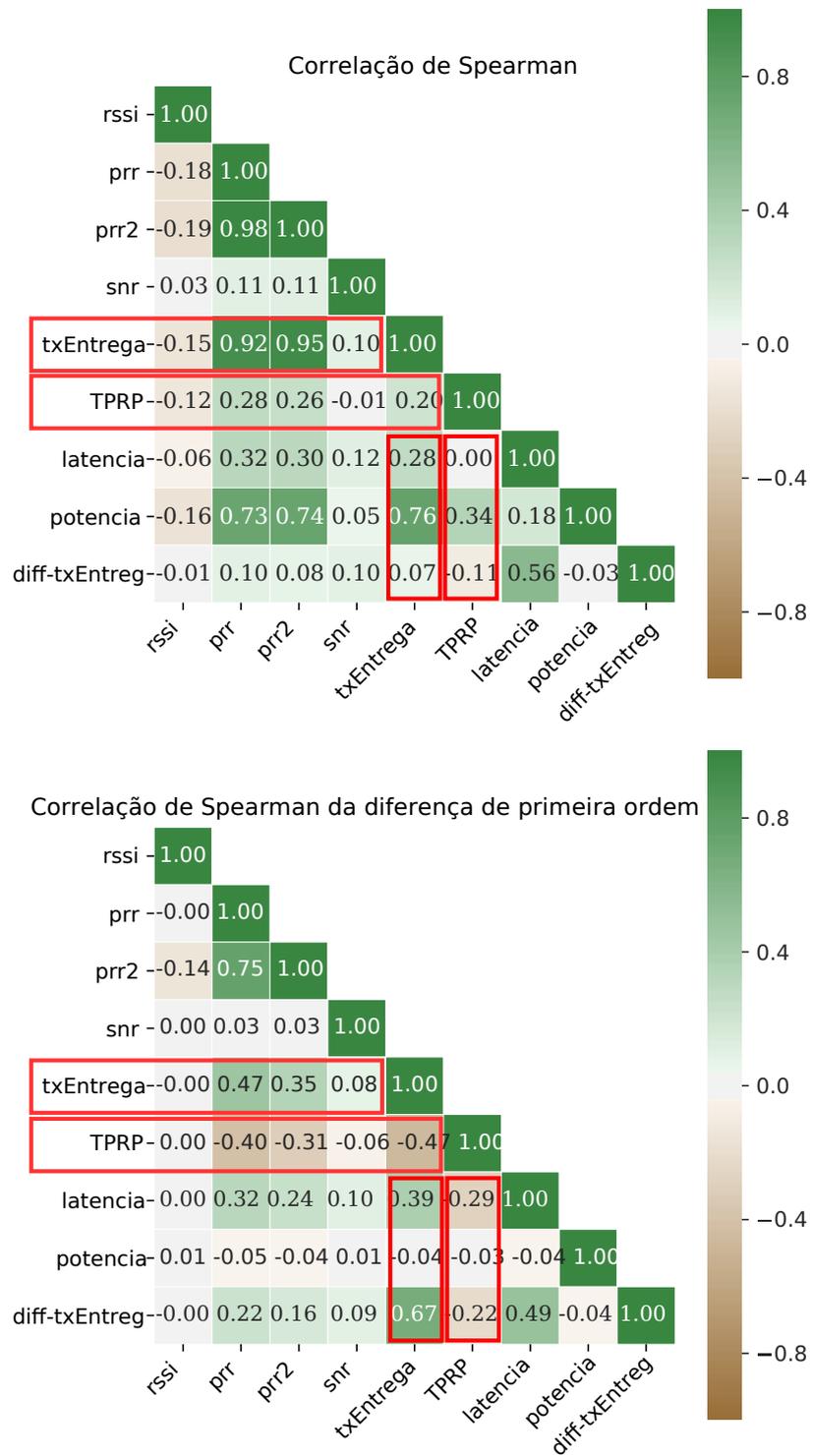


Figura 5.6. Correlação de Spearman para 8 features

Tabela 5.2. Matriz de correlação de Pearson

	rsssi	prrr	snr	taxadeEntrega	TPRP
rsssi	–	-0.6986	-0.3431	-0.7094	0.1619
prrr	-0.6986	–	0.2690	0.9615	-0.2300
snr	-0.3431	0.2690	–	0.2624	-0.0641
taxadeEntrega	-0.7094	0.9615	0.2624	–	-0.2281
TPRP	0.1619	-0.2300	-0.0641	-0.2281	–

de recursos com magnitudes igual ou próximo a zero ($< 0, 1$) foram descartadas. As melhores combinações (isto é, valores próximos a 1) foram as seguintes:

Inputs: rsssi, prrr, snr, latência

Target: taxadeEntrega

e

Inputs: rsssi, prrr, snr, latência

Target: TPRP

Os resultados apresentados no mapa de calor da Correlação de Spearman, selecionando os recursos que apresentam mais valores próximos a 1, nos permitiram construir o algoritmo de aprendizado de máquina com *taxadeEntrega* e *TPRP* como destino e *rsssi, prrr, snr, latência* como entradas, com o primeiro apresentando mais características correlacionadas, enquanto a combinação com o TPRP como alvo apresenta características com menor correlação.

5.3.3 Resultados do treinamento

Para identificar o algoritmo de *Machine Learning* a ser utilizado no LQM, foi realizado um teste com oito algoritmos e coletados os valores de precisão e dos desvios-padrão. Esses algoritmos foram escolhidos porque possuem características que podem ser úteis para o processamento dos dados; por exemplo, o algoritmo de regressão linear⁴ [Cármenes, 1991] foi escolhido para avaliar as condições sob as quais os recursos de entrada são relevantes para a saída, mas enquanto não forem altamente correlacionados entre si; o SVR⁵ [Chang & Lin, 2011] foi escolhido por sua capacidade de capturar relacionamentos muito mais complexos entre os dados, e LinearSVR⁵ [Chang & Lin, 2011] e NuSVR⁵ [Chang & Lin, 2011] são semelhantes ao SVR, mas usam parâmetros específicos de kernel. O KNN⁶ [Guo et al., 2003] é um algoritmo não paramétrico (ou seja, não há suposição para a distribuição de dados subjacente, recurso importante que o fez ser avaliado. A regressão em árvore

⁴https://scikit-learn.org/stable/modules/linear_model.html

⁵<https://scikit-learn.org/stable/modules/svm.html>

⁶<https://scikit-learn.org/stable/modules/neighbors.html>

de decisão⁷ [Dumont et al., 2009] foi escolhida para avaliação por ser um algoritmo rápido e clássico, e a rede Neural (NNET)⁸ [Rumelhart et al., 1986] também foi escolhida por ser um método clássico no *Machine Learning*. O algoritmo Gradient Boosting para regressão⁹ [Ke et al., 2017] é um novo método de conversão de aprendizes fracos em fortes, motivo pelo qual foi escolhido para avaliação. O módulo Python do scikit-learning Pedregosa et al. [2011] foi usado para treinamento e execução de algoritmos de ML. As seguintes técnicas e métricas de avaliação foram utilizadas.

Cross-validation com Shuffle Split: esta técnica é utilizada para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados. O conceito central desta técnica é o particionamento do conjunto de dados em subconjuntos mutuamente exclusivos seguido do uso de alguns destes subconjuntos para a estimação dos parâmetros do modelo, sendo os subconjuntos restantes empregados na validação do modelo. O parâmetro *shuffle split* irá gerar um número definido pelo usuário de divisões independentes do conjunto de dados embaralhados, e na sequência serão divididas em um par de conjuntos de treinamento e de teste.

Cross-validation com kFold: este método consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos do tamanhos iguais, então um subconjunto é escolhido para teste e os $k-1$ restantes são utilizados para avaliar a precisão do modelo. Este processo é realizado k vezes alternando de forma circular o subconjunto de teste.

Divisão 80% - 20%: esta técnica utiliza uma proporção fixa, neste caso, 80% - 20%, para realizar a divisão entre conjuntos de treinamento e de testes para o modelo. Tem como vantagem ser mais rápida que uma técnica que utiliza *cross-validation*.

RMSE (Root Mean Squared Error): é uma métrica utilizada para medir a acurácia de modelos preditivos. Indica a raiz quadrática média dos erros entre valores observados e as previsões. Quanto menor o valor, melhor.

MAE (Mean absolute error): esta métrica mede a magnitude média dos erros em um conjunto de previsões, sem considerar sua direção. É a média sobre a

⁷<https://scikit-learn.org/stable/modules/tree.html>

⁸https://scikit-learn.org/stable/modules/neural_networks_supervised.html

⁹<https://scikit-learn.org/stable/modules/ensemble.html>

amostra de teste das diferenças absolutas entre a previsão e a observação real, onde todas as diferenças individuais têm peso igual.

Explained Variance: esta métrica é utilizada para medir a discrepância entre um modelo e os dados reais. Em outras palavras, é a parte da variância total do modelo que é explicada por fatores que estão realmente presentes e não é devido à variância do erro. Quanto mais próximo os resultados forem de 1, melhor.

Os resultados da Tabela 5.3 foram obtidos para as *features* *rss*, *pr*, *snr*, *TPRP*, *latencia* e *target taxadeEntrega*, enquanto os resultados na tabela 5.4 considera *TPRP* como *target*.

Tabela 5.3. Desempenho dos algoritmos de *Machine Learning* - *target*: *taxadeEntrega*

	Cross val. (Shuffle split=5)	Cross val. (Kfold=5)	80/20% RMSE	80/20% Explained variance score ¹⁰	80/20% MAE ¹⁰
Linear Regression	0.93 (+/- 0.00)	0.93 (+/- 0.00)	8.635	0.927	5.697
SVR	0.92 (+/- 0.00)	0.92 (+/- 0.00)	9.049	0.921	4.932
LinearSVR	0.92 (+/- 0.02)	0.92 (+/- 0.02)	9.201	0.924	5.537
NuSVR	0.78 (+/- 0.00)	0.78 (+/- 0.00)	14.865	0.812	13.430
KNN	0.94 (+/- 0.00)	0.94 (+/- 0.00)	7.741	0.942	4.080
Decision tree Regression	0.97 (+/- 0.00)	0.97 (+/- 0.00)	5.693	0.968	2.331
NNet	0.96 (+/- 0.02)	0.96 (+/- 0.00)	6.344	0.962	4.053
GBE	0.97 (+/- 0.00)	0.97 (+/- 0.00)	5.811	0.967	3.257

A métrica R^2 foi utilizada para determinar a precisão do método de validação cruzada (*cross validation*). O coeficiente R^2 é definido como $(1 - u/v)$, onde u é a soma residual dos quadrados e v é a soma total dos quadrados. Considerando os resultados apresentados na Tabela 5.3, os dois métodos de validação cruzada (*shuffle split* e *Kfold*) indicam os algoritmos *Decision Tree Regressor* e *Gradient Boost* como os melhores em termos de acurácia, com discreta vantagem para o primeiro, que também é reforçado pelos indicadores RMSE (*Root-mean-square square*), EVS

¹⁰A principal diferença entre as métricas R^2 e EVS (*Explained Variance Score*) é que este considera o resíduo médio em sua computação. MAE significa *Mean Absolute Error*

Tabela 5.4. Desempenho dos algoritmos de *Machine Learning* - *target*: TPRP

	Cross val. (Shuffle split=5)	Cross val. (Kfold=5)	80/20% RMSE	80/20% Explained variance score	80/20% MAE
Linear Regression	0.05 (+/- 0.01)	0.05 (+/- 0.01)	224.631	0.055	117.535
SVR	-0.09 (+/- 0.01)	-0.09 (+/- 0.01)	242.387	-0.007	71.174
LinearSVR	-0.09 (+/- 0.02)	-0.09 (+/- 0.02)	242.809	-0.003	73.833
NuSVR	-1.41 (+/- 0.08)	-1.41 (+/- 0.08)	357.767	-0.001	345.049
KNN	-0.25 (+/- 0.07)	-0.25 (+/- 0.07)	257.817	-0.243	99.875
Decision tree Regression	-0.49 (+/- 0.12)	-0.48 (+/- 0.13)	276.974	-0.435	91.007
NNet	0.06 (+/- 0.01)	0.07 (+/- 0.01)	223.057	0.069	112.920
GBE	0.21 (+/- 0.02)	0.21 (+/- 0.02)	205.458	0.210	91.572

e MAE (*Mean Absolute Error*). Devido à sua maior precisão, o algoritmo *Decision Tree Regressor* foi escolhido para a operação no LQM. Para o algoritmo de ML selecionado, foram utilizados os valores especificados na Tabela 5.5 para os hiperparâmetros, obtidos em Pedregosa et al. [2011] como configuração padrão aplicável em diversos cenários.

Tabela 5.5. Hiperparâmetros utilizados na ML

Itens	Valor
Presort	False
splitter	best
min_impurity_decrease	0.0
max_leaf_nodes	none
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
criterion	mse
random_state	none
min_impurity_split	none
max_features	none
max_depth	none

Os resultados entregues pelos algoritmos de ML quando utilizamos o TPRP como alvo (*target*), apresentados na Tabela 5.4, são inferiores e os algoritmos que

se sobressaem são GBE e *Linear regression*, a depender também das métricas de avaliação. Também foi perceptível a diferença de tempo de treinamento de alguns algoritmos ao utilizar *recursos* correlacionados (4 min 36 seg) e ao usar características com menor correlação (54 min 26 seg), executados em um Computador Intel Core i5-8250u com 8 Gb de RAM e SSD de 120 Gb, com sistema Linux Mint 19. O tempo de treinamento do algoritmo de *Decision tree regression* foi de 2,02 segundos e o tempo de execução aprox. 0,2 seg. Os longos períodos de treinamento para redes neurais enfatizam a não escolha de técnicas de aprendizagem profunda. A Tabela 5.6 apresenta o tempo de treinamento para todos os 8 algoritmos de ML avaliados. Utilizando o *plugin* Python *memory-profiler*¹¹, identificamos um pico de consumo de memória de 269 MB durante o treinamento.

Tabela 5.6. Tempo de treinamento (*cross validation* e *shuffle split*)

Algoritmo	Tempo de treinamento	Algoritmo	Tempo de treinamento
Linear Reg.	0.33 sec.	SVR	27 min 34s
KNN	0.59 sec.	DT Reg.	2.02 sec.
LinSVR	40.8 sec.	NNET	3 min 58 s
NuSVR	6 min 54 s	GBE	10.94 sec

Em seguida, avaliamos o algoritmo ML mais eficaz em uma plataforma experimental. Isso é importante, pois o LQM é implementado em nós com restrições de recursos. Adotando o equipamento Raspberry Modelo Pi 3 B da bancada do testbed Futebol, o tempo de treinamento do algoritmo de regressão em árvore de decisão foi de 3,2 segundos e consumiu 48,5 MB de memória, identificados utilizando o *plug-in* Python *memory-profiler*.

5.3.4 Identificando *Concept Drift*

Para identificar a ocorrência de *concept drift*, em razão desta etapa de definição dos algoritmos ser realizada *offline* e não ter restrições de tempo de processamento nem de consumos energéticos e de memória, foram utilizados os algoritmos ADWIN [Bifet & Gavalda, 2007] e CUSUM [Basseville et al., 1993]. Durante a execução *online*, o LQM adota o ADWIN. O valor padrão 0,01 foi usado para o Adwin Delta, e os algoritmos básicos podem ser encontrados no repositório do PyAdwin¹². O algoritmo CUSUM foi executado com os parâmetros nível de confiança = 90 e *offset* = 0, e

¹¹<https://pypi.org/project/memory-profiler/>

¹²<https://github.com/rsdevigo/pyAdwin>

a implementação foi baseada na implementação disponível em ¹³ devido ao fornecimento de uma explicação detalhada das abordagens de CUSUM.

Das 10.000 amostras, identificamos 19 ocorrências de desvio ou cortes (0,19 %), usando ADWIN, enquanto 3 cortes (0,03 %) foram obtidos usando o CUSUM Baseville et al. [1993], sendo um indicativo de que os dados apresentam *concept drift*, sugerindo, portanto, que a abordagem adotada deve considerar o CD.

Diante da detecção de CD e considerando um sistema que possui restrições de memória disponível, foi adotada a estratégia de retreinar utilizando pequenas bases e algoritmos que processam o treinamento rapidamente, como o Algoritmo da Árvore de Decisão, conforme descrito na subseção anterior.

5.4 Conclusão

Este capítulo apresentou o LQM, um estimador de qualidade de enlace de múltiplas camadas que aplica técnicas de aprendizagem de máquina com detecção de *concept drift*. O estimador LQM combina as métricas RSSI, SNR, PRR, latência e TPRP para produzir uma estimativa acurada da qualidade do enlace. Foi utilizada uma plataforma de teste com experimentos reais para o processo de desenvolvimento.

¹³<http://www.variation.com/cpa/tech/changepoint.html>

Capítulo 6

Resultados e Análise

Este capítulo apresenta os resultados obtidos com a avaliação experimental dos estimadores propostos utilizando equipamentos USRP (*Universal Software Radio Peripheral*). Inicialmente, na primeira seção, são descritas as métricas de avaliação comuns a ambos estimadores propostos. Na segunda seção são apresentados os resultados do estimador PRR² e na terceira seção estão apresentados os resultados do estimador LQM.

6.1 Avaliação

Como o cenário de avaliação considera uma comunicação M2M que requer comunicação confiável e de baixa potência, as seguintes métricas são consideradas durante a avaliação dos estimadores PRR² e LQM:

Taxa de entrega de pacotes (*Packet delivery ratio* - PDR): é a razão de pacotes recebidos com sucesso em relação ao total enviado. Contribui para indicar a qualidade da comunicação percebida pelo usuário.

Índice do ganho de transmissão (*Transmission gain index* - TGI): indica a quantidade de energia gasta por transmissão (um ganho mais alto requer uma maior potência de transmissão, o que aumenta o consumo energético). O TGI é o valor do ganho em dBm usado pelo equipamento USRP. Aumenta ou atenua a potência de transmissão, de acordo com cada modelo do USRP. No modelo B210, o índice de ganho varia de 0 a 89 dB e, no modelo B100, o índice de ganho varia de 0 a 35 dB.

Pacotes transmitidos por pacotes recebidos (*Transmitted Packets per Received Packets* - TPRP): representa a relação entre pacotes enviados, incluindo retransmissões, e pacotes recebidos. Pode ser uma indicação de quão eficiente foi o processo de entrega, dado que para um pacote recebido, em algumas situações, pode ser necessário mais de uma transmissão do mesmo pacote.

6.2 Resultados dos experimentos com PRR²

Esta seção apresenta a avaliação experimental do estimador PRR² proposto utilizando *hardware* programável. Inicialmente é descrita a implementação do PRR², seguida da descrição dos cenários experimentais, e por fim são apresentados os resultados obtidos.

6.2.1 Implementação do PRR² no USRP

Para atingir os objetivos deste trabalho, implementamos o PRR² em equipamentos USRP modelos B210 com programação no ambiente GNU Radio utilizando a linguagem Python. Em virtude do protocolo IEEE 802.15.4 ser uma opção comumente adotada para comunicação máquina a máquina, esse foi o protocolo adotado para a realização dos experimentos.

Os USRPs executam os módulos disponíveis na biblioteca do GNU Radio ou aqueles desenvolvidos para um propósito específico. Implementamos os módulos para obtenção das métricas de entrada (RSSI, *ack* e perda de pacote), permitindo estimar a qualidade do enlace de comunicação. Em seguida implementamos o módulo responsável pelas fases de suavização, normalização e computação do PRR². Também implementamos um módulo que permite que a potência de transmissão seja modificada dinamicamente, conforme descrito na próxima seção.

Outros membros do nosso grupo de pesquisa realizaram modificações na pilha de protocolo do IEEE 802.15.4, originalmente desenvolvida pelos autores de Bloessl et al. [2013]. A implementação original não fornece uma camada funcional MAC, isto é, não há detecção de portadora e nem retransmissões. Eles também desenvolveram o mecanismo de detecção de portadora, bem como o algoritmo binário exponencial *Back-off* para retransmissões. Com isso, informações de *ack* e de perda de pacotes transmitidos também passaram a ser providas.

6.2.2 Descrição do algoritmo de controle de potência

Para a realização e avaliação dos experimentos, foi necessário implementar um módulo de controle de potência da transmissão no GNU Radio e integrá-lo ao módulo do estimador proposto. A Figura 6.1 apresenta essa integração, ilustrando que a entrada do módulo de potência é a estimativa produzida pelo PRR², e a saída é o valor da potência de transmissão atualizado.

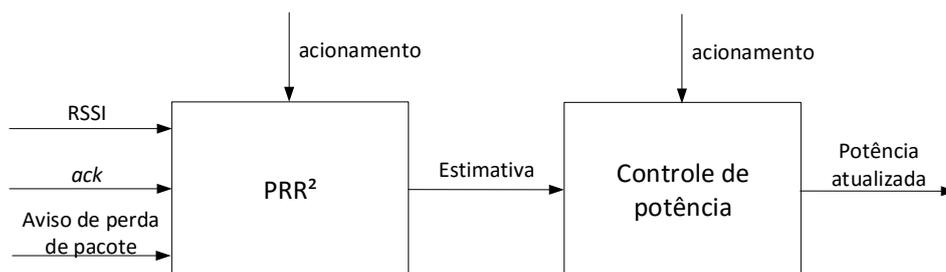


Figura 6.1. Entrada e saída do módulo de controle de potência

Visto que o foco do trabalho é o estimador de qualidade do enlace, o funcionamento do controle de potência é o mais simples possível. Como ele foi desenvolvido especificamente para os experimentos com o USRP B210, as calibrações apresentadas a seguir podem necessitar de ajustes para funcionamento adequado em outros modelos de USRP. O valor de saída do estimador é utilizado para determinar o *gain index* do equipamento transmissor. *Gain index* é um valor expresso em dB que é utilizado para intensificar ou atenuar a potência de transmissão, de acordo com cada modelo de USRP. A Figura 6.2 ilustra o mapeamento entre os valores estimados de qualidade do enlace e os valores atribuídos para o ganho de potência da transmissão.

Os valores estimados de qualidade do enlace variam de 0 a 1, enquanto os valores de *gain index* variam de 21 a 89 dB. Foi constatado empiricamente que uma transmissão com ganho inferior a 21 dB não gera transmissões com sucesso, motivo pelo qual o menor valor de *gain index* adotado é 21 dB. O valor do *ganho* de 89 dB refere-se ao ganho máximo de potência de transmissão aceito pelo USRP modelo B210.

O cálculo da potência, utilizado no módulo controle de potência, pode ser encontrado na Equação 6.1 a seguir.

$$potencia = [72 * (1 - estimativa)] + 21 \quad (6.1)$$

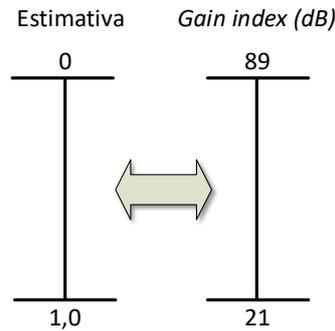


Figura 6.2. Relação entre os valores estimados e o ganho de potência da transmissão

Para todos os estimadores de qualidade abordados experimentalmente neste trabalho, o módulo de controle de potência atua da mesma forma: recebe como entrada um valor estimado da qualidade do enlace, de 0 a 1, computando então a saída para a potência de transmissão, por meio do *gain index*. Para isso, os estimadores de qualidade de enlace foram normalizados para produzir saída de 0 a 1.

6.2.3 Cenários de experimentação

O estimador PRR² proposto foi avaliado em um *testbed*, utilizando rádios definidos por *software*. Foi projetado um cenário experimental que retrata a transmissão de dados entre dois dispositivos de baixa potência, por exemplo, um monitor de frequência cardíaca enviando pacotes para um servidor ou aparelho coletor de dados. Neste cenário, o principal objetivo é transmitir dados de forma confiável, consumindo a menor quantidade de energia e demandando menos retransmissões. Para isso, foram utilizadas as estimativas de enlace calculadas pelo PRR² para ajustar dinamicamente a potência de transmissão durante o envio dos pacotes.

A Figura 6.3 apresenta a disposição dos equipamentos durante um experimento real com o PRR², exibindo o transmissor USRP TX, local onde o PRR² atua, o receptor nomeado USRP RX, e um terceiro equipamento responsável por adicionar ruído sintético na mesma frequência de operação dos demais equipamentos.

Os experimentos foram realizados utilizando o *hardware* Ettus USRP B210 [Research, 2016] do *testbed* FUTEBOL UFMG ¹, que fornece um transceptor de canal duplo operando na faixa de frequência de 70 MHz - 6 GHz e suporte a MIMO (*Multiple Input Multiple Output*). Neste trabalho, a frequência de operação é 2,4 GHz.

¹<http://www.futebol.dcc.ufmg.br/>

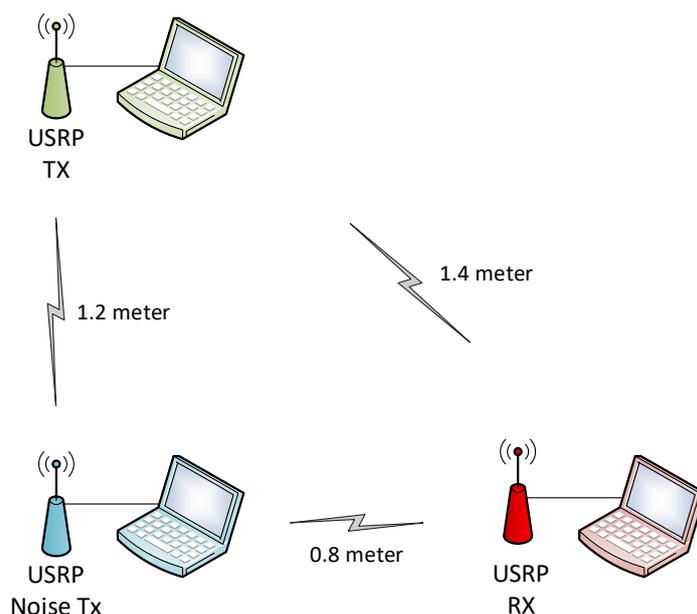


Figura 6.3. Disposição dos USRPs durante um experimento com o PRR^2

Nos experimentos realizados, o PRR^2 emprega uma janela de 40 observações para *shortPRR*, enquanto o *longPRR* é calculado sobre 80 observações. Esses valores foram definidos empiricamente. Um dos objetivos do experimento é comparar o estimador proposto com um que também seja simples e rápido. Para isso, PRR^2 é comparado com KLE, um estimador que emprega filtros de Kalman sobre a métrica RSSI [Senel et al., 2007].

Dois cenários são analisados: no *primeiro cenário* objetiva-se determinar a contribuição de cada fator do PRR^2 . Neste cenário foram enviados 80 pacotes, sem contabilizar as retransmissões, e foram realizadas 5 repetições de todo o processo de execução. No *segundo cenário* objetiva-se fazer uma comparação com estimador da literatura. Foi realizada a comparação do PRR^2 com o estimador KLE, com o envio de 200 pacotes, sem contabilizar as retransmissões, e foram realizadas 3 repetições. Os principais resultados são discutidos na próxima seção.

6.2.4 Resultados

6.2.4.1 Primeiro cenário: contribuição dos estimadores

Para identificar como cada parte do PRR^2 contribui para o desempenho final, dividimos o PRR^2 em 5 estimadores menores: (1) RSSI com média móvel exponencial,

visando avaliar a participação do RSSI nos resultados; (2) PRR com uma janela de observação de tamanho 40, objetivando verificar a participação da janela curta; (3) PRR com uma janela de observação de tamanho 80, objetivando analisar a participação da janela longa; (4) PRR combinado com a estimativa de RSSI, visando avaliar a participação da combinação do PRR e RSSI nos resultados; (5) e PRR com janelas longas e curtas, mas sem estimativa de RSSI, visando analisar a participação de ambas janelas, mas sem o RSSI. As legendas desses estimadores nos resultados são, respectivamente: (1) RSSI, (2) PRR w 40, (3) PRR w 80, (4) PRR+RSI, e (5) 2 PRR.

A Figura 6.4 apresenta as taxas de entrega para a comparação anteriormente mencionada. Os resultados estão representados graficamente com um intervalo de confiança de 95 %. A contribuição é menor no RSSI, enquanto uma maior contribuição é observada nos estimadores PRR w 40 e PRR w 80. Isto se deve ao fato de que o PRR é uma métrica mais robusta, uma vez que mede a qualidade do enlace em ambas as direções devido à sua dependência de pacotes ACK. Em seguida, mesclando as estimativas de duas camadas (PRR + RSSI), aumenta-se ligeiramente o desempenho, um efeito similar ao obtido com o 2 PRR . Finalmente, o PRR², que utiliza duas janelas PRR e RSSI, aumenta a taxa de entrega em 3% quando comparado a um estimador que combina as duas camadas (RSSI e PRR). Assim, o uso de duas janelas PRR suaviza as variações devido a variações momentâneas da qualidade do enlace. Ressaltamos que os intervalos de confiança (IC) de PRR² e PRR + RSSI não se sobrepõem, como pode ser visto na Tabela 6.1.

Tabela 6.1. Médias e intervalos de confiança da taxa de entrega

	RSSI	PRR w 40	PRR w 80	PRR + RSSI	2 PRR	PRR ²
IC+	18,30%	52,35%	52,20%	53,99%	54,23%	57,98%
Média	13,50%	51,75%	51,00%	53,50%	53,25%	56,25%
IC-	8,70%	51,15%	49,80%	53,01%	52,27%	54,52%

Também são observadas reduções na quantidade de retransmissões por pacote recebido quando comparado aos estimadores de RSSI, como mostrado na Figura 6.5: enquanto o RSSI apresenta uma média de 50 retransmissões por pacote recebido, os estimadores que utilizam a métrica PRR apresentaram, no pior dos casos, uma média de 8,25 retransmissões por pacote recebido. Um número menor de retransmissões também é importante para uma rede confiável, pois reduz os atrasos na entrega de pacotes. Além disso, devemos notar que estimadores mais precisos também reduzem os intervalos de confiança. Isso é traduzido para o usuário como menor *jitter*, que é importante em aplicações de controle em tempo real.

As melhorias de desempenho do PRR² foram obtidas com o aumento do *Gain index* de transmissão, como mostrado na Figura 6.6. O PRR² utilizou um ganho médio de 56,6 dB, ao passo que RSSI utilizou 52,8 dB. Os reduzidos intervalos de confiança de PRR² quando comparados com as outras configurações de PRR também mostram que o uso de duas janelas PRR reduz a variabilidade na potência de transmissão, o que também é importante para um comportamento de enlace estável.

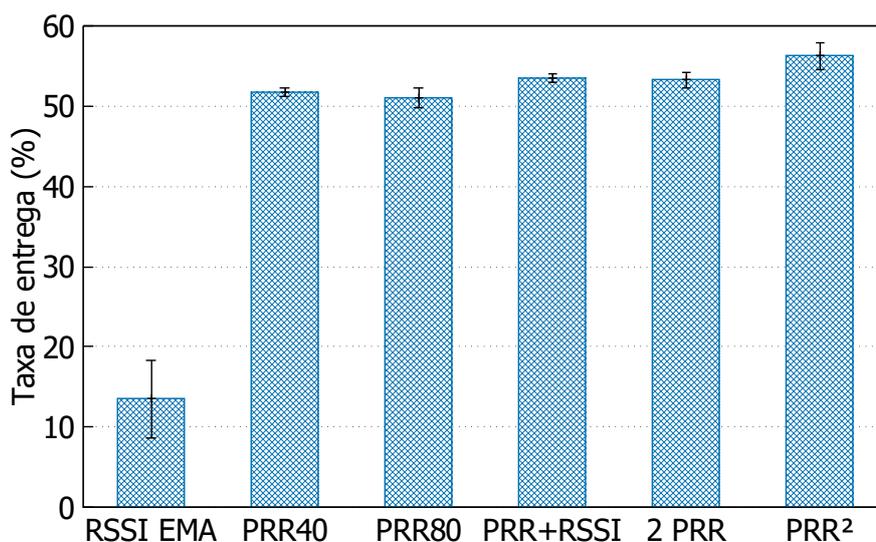


Figura 6.4. Comparações para taxa de entrega

6.2.4.2 Segundo cenário: comparação com estado da arte

Neste *segundo cenário* é realizada uma comparação do PRR² com o estimador KLE. Subdivide-se em duas configurações: na *primeira configuração*, o intervalo entre transmissões de mensagens variou de 50 ms a 150 ms com o ganho do ruído artificial fixado em 66 dB. Na *segunda configuração*, o intervalo entre as mensagens foi fixado em 150 ms e o ganho do ruído artificial variou entre 74 dB e 80 dB.

1ª configuração: Intervalo de transmissões variável e ruído fixo A Figura 6.7 exibe a taxa de entrega dos pacotes obtida utilizando os estimadores RSSI com filtro de Kalman, PRR e PRR². O estimador PRR² possibilitou maior taxa de entrega nos três intervalos de transmissão observados; contudo, nos intervalos de transmissão de 50 ms e 100 ms, houve uma sobreposição dos intervalos de confiança projetados, o que não permite afirmar estatisticamente qual foi o melhor estimador nestes casos.

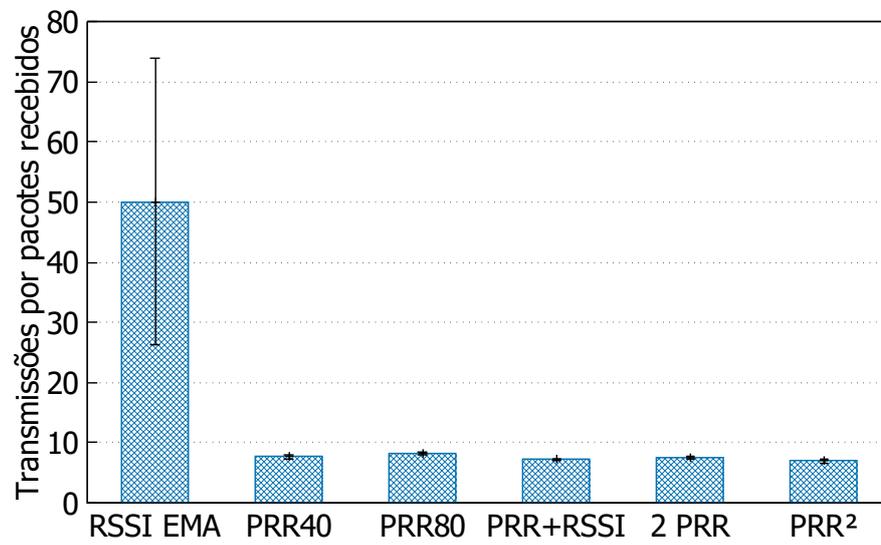


Figura 6.5. Transmissões por pacotes recebidos

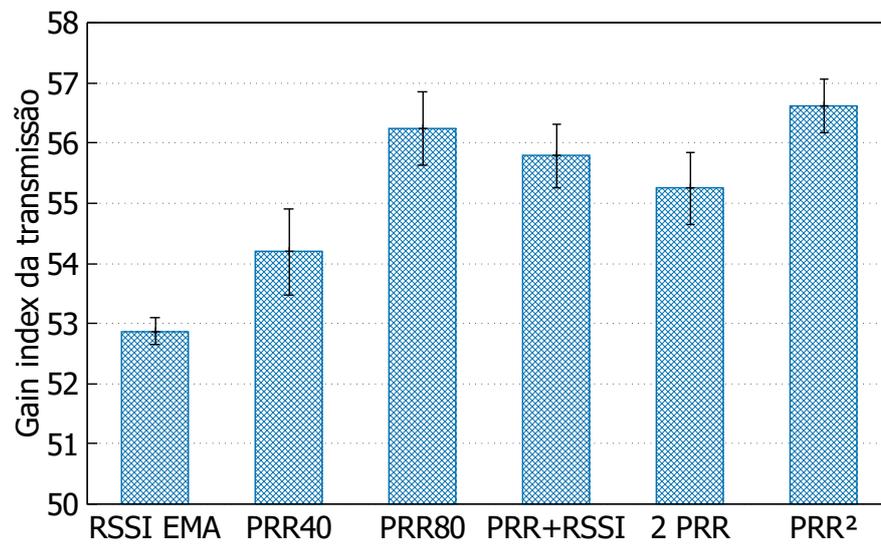


Figura 6.6. Gain index da transmissão

Quando observado o intervalo de transmissão de 150 ms, nota-se que mesmo considerando os intervalos de confiança, o estimador proposto PRR^2 possibilitou maior taxa de entrega que os demais estimadores, proporcionando uma melhoria de 10,3% na taxa de entrega, em média, com relação ao KLE. Além disso, os intervalos de confiança mais curtos obtidos pelo PRR^2 nos três intervalos de transmissão observados são interpretados como uma maior estabilidade nas estimativas, em comparação aos demais estimadores, que projetaram longos intervalos de confiança.

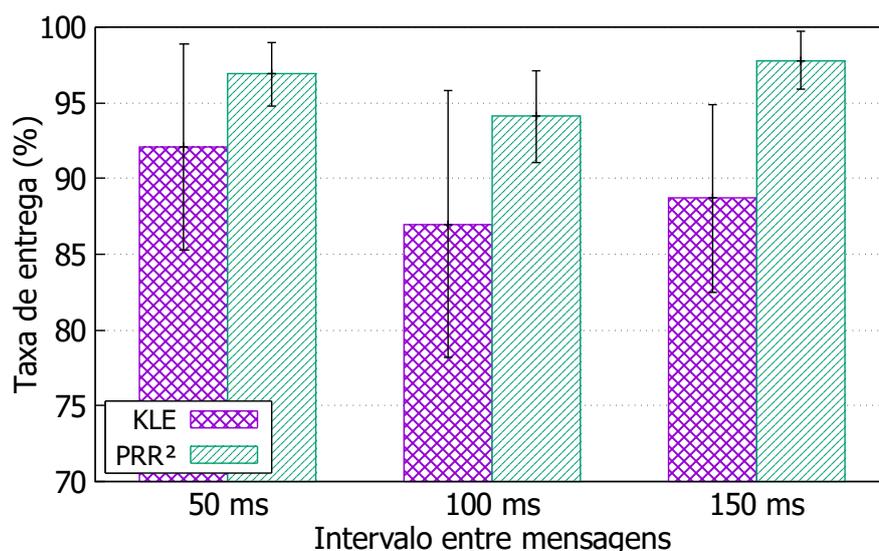


Figura 6.7. Taxa de entrega

O PRR^2 também contribuiu para reduzir, em média, 63,1% a quantidade de transmissões por pacotes recebidos com sucesso, quando observado no intervalo de transmissão de 150 ms, conforme mostrado na Figura 6.8. No intervalo de transmissão de 50 ms o PRR^2 também obteve uma melhor média, porém os intervalos de confiança sobrepostos e com a média do PRR^2 contida no intervalo de confiança do KLE impedem uma conclusão estatisticamente precisa. Para isso, é necessário realizar o *teste-t de Student* a fim de validar a hipótese de que as médias são estatisticamente diferentes (hipótese alternativa). A hipótese nula é de que as médias do PRR^2 e KLE são estatisticamente iguais. Utilizamos o *teste-t* bi-caudal e obtivemos o *p-value* de 0,39. Como o *p-value* é maior que o nível de significância 0,05, para um nível de confiança de 95%, a hipótese nula é aceita, o que nos permite concluir que não há diferença significativa entre as duas médias para esse nível de confiança e 50 ms de intervalo de transmissão. Para o intervalo de transmissão de 100 ms,

utilizando o *teste-t de Student*, obtivemos o *p-value* de *0,28*, também confirmando a hipótese nula.

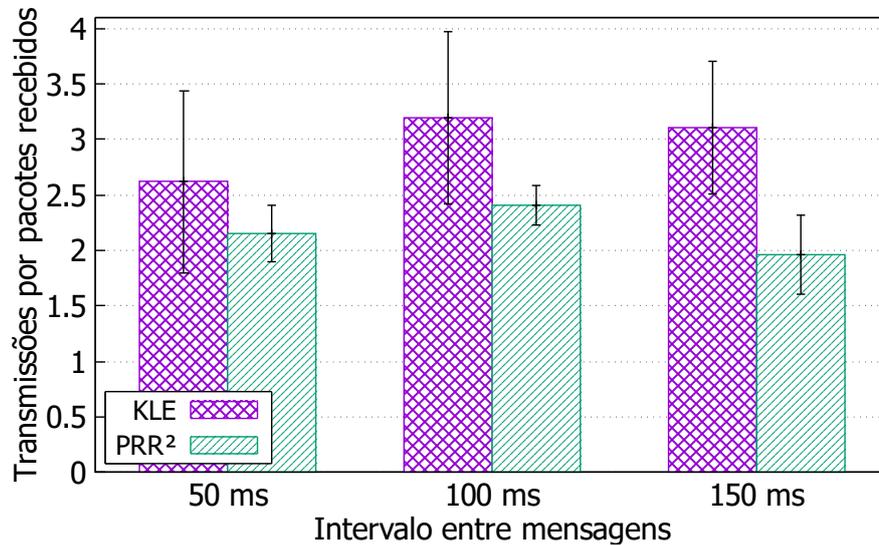


Figura 6.8. Número de retransmissões por pacote recebido

Apesar do melhor resultado na taxa de entrega quando o intervalo de transmissão é de 150 ms (Figura 6.4), a diferença de potência de transmissão entre o RSSI com Kalman e o PRR² é de apenas 0,07%, em média, tal como apresentado na Figura 6.9. O *teste-t de Student* resultou em um *p-value* de *0,957*, indicando que estatisticamente não há diferenças significativas entre as médias de PRR² e KLE. Dessa forma, observamos que o PRR² proporciona uma melhor taxa de entrega sem utilizar maior potência de transmissão.

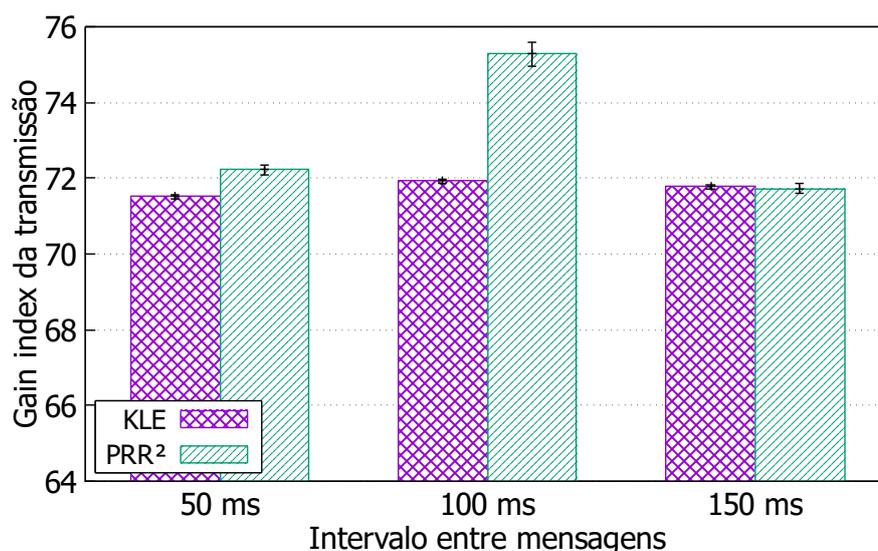


Figura 6.9. Ganho na transmissão

2ª configuração: Intervalo de transmissões fixo e ruído variável Os experimentos realizados com esta segunda configuração objetivam avaliar o efeito do ruído nas estimativas proporcionadas pelos algoritmos. Os valores do *Gain index* variam entre 74 dB, 77 dB e 80 dB. O intervalo de transmissões ficou fixo em 150 ms.

A Figura 6.10 mostra a taxa de entrega quando o intervalo de transmissão está fixo em 150 ms e o ganho da potência do ruído varia entre 74, 77 e 80 dB. Os melhores resultados, conforme esperado, foram obtidos com um ruído de menor potência ($G=74$). Neste nível de potência de ruído, o PRR² proporcionou uma taxa de entrega de 98,50%, enquanto o KLE entregou 77,75%, em média. O intervalo de confiança do PRR² não ficou sobreposto a nenhum outro intervalo, permitindo inferir que, neste cenário, o PRR² foi o que apresentou a melhor eficiência na entrega dos pacotes. Análise similar pode ser obtida com o ruído de 77 dB, com conclusão também indicando uma significativa diferença estatística para a taxa de entrega a favor do PRR². Com o ruído de 80 dB, os resultados apresentam sobreposição nos intervalos de confiança, exigindo o *teste-t de Student* para confirmação das conclusões. A hipótese nula é de que as médias do PRR² e KLE são estatisticamente iguais. Utilizamos o *teste-t* bi-caudal e obtivemos o *p-value* de 0,098. Como o *p-value* é maior que o nível de significância 0,05, para um nível de confiança de 95%, a hipótese nula é aceita, o que nos permite concluir que não há diferença significativa entre as duas médias para esse nível de confiança. Por sua vez, adotando-se o nível de confiança de 90%, a hipótese nula pode ser rejeitada, indicando que neste nível

de confiança há diferenças significativas entre as médias de PRR^2 e KLE.

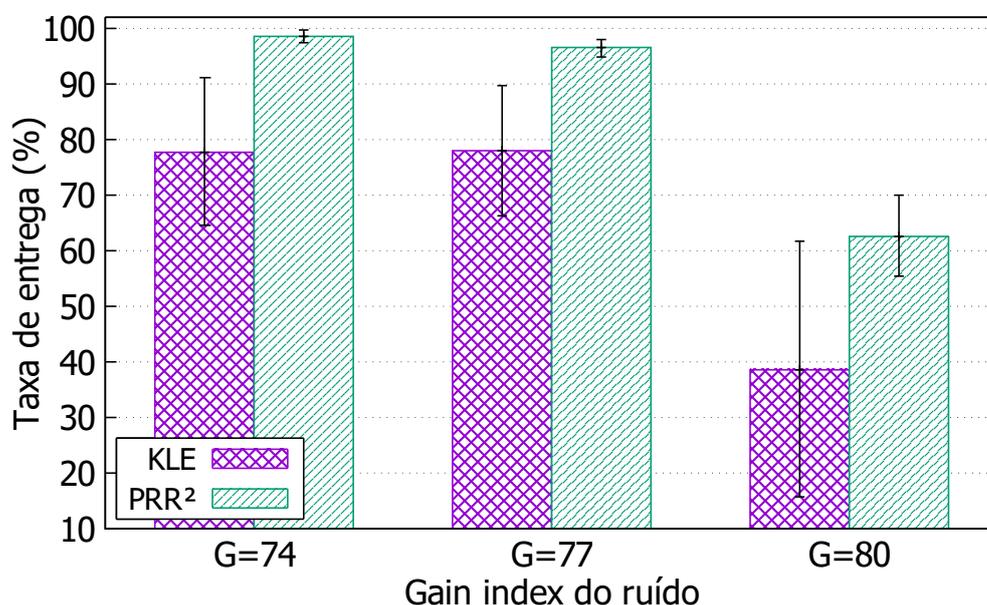


Figura 6.10. Taxa de entrega no segundo cenário

A Figura 6.11 mostra a quantidade de transmissões por pacote recebido com o intervalo de transmissão fixo em 150 ms. Com o ruído de 74 dB, o PRR^2 propiciou uma redução de 54,7% na razão de transmissões por pacotes recebidos, em média, em comparação ao método KLE. Com o ruído de 77 dB, o PRR^2 também se apresenta como o melhor estimador, exigindo menos retransmissões. Com o ruído de 80 dB, os resultados apresentaram intervalos de confiança longos, com sobreposição entre eles. Utilizamos o teste-*t* bi-caudal e obtivemos o *p-value* de 0,14 entre as médias de PRR^2 e KLE e *p-value* de 0,39 entre as médias de PRR^2 e PRR. Dessa forma, não podemos rejeitar a hipótese de que não há diferença significativa entre as médias analisadas.

A Figura 6.12 apresenta a potência de transmissão utilizada pelos estimadores. Com a fonte de ruído utilizando ganho de potência de transmissão 74 e 77, o PRR^2 apresentou melhor desempenho, utilizando menor potência média de transmissão em comparação ao estimador KLE. A situação com a fonte de ruído utilizando ganho de potência de transmissão de 80 dB mostra o PRR^2 empregando 13,7% mais potência de transmissão do que o método KLE para prover uma melhoria de 62,1% na taxa de entrega de pacotes (Figura 6.10).

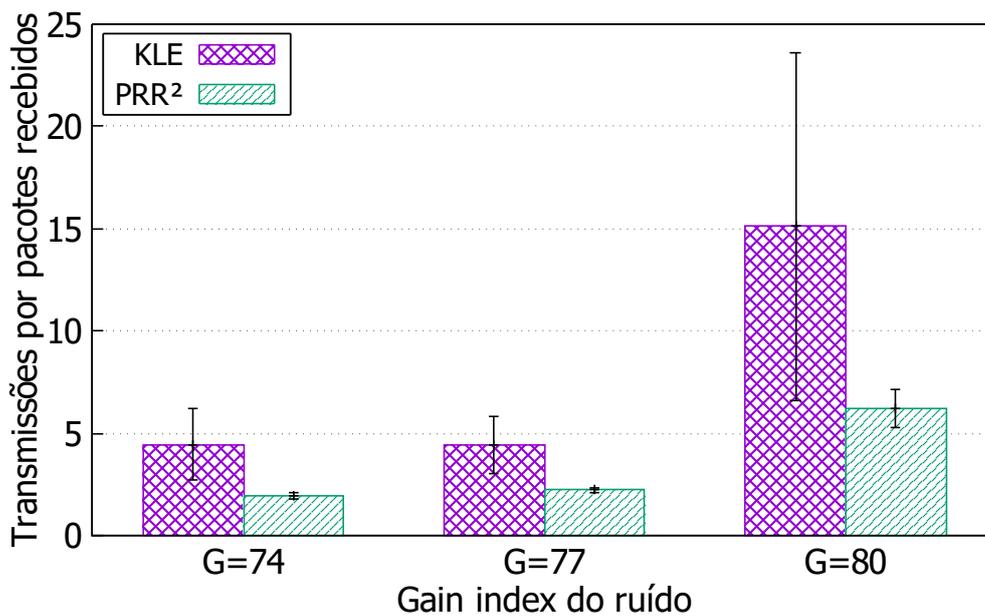


Figura 6.11. Quantidade de retransmissões por pacote recebidos no segundo cenário

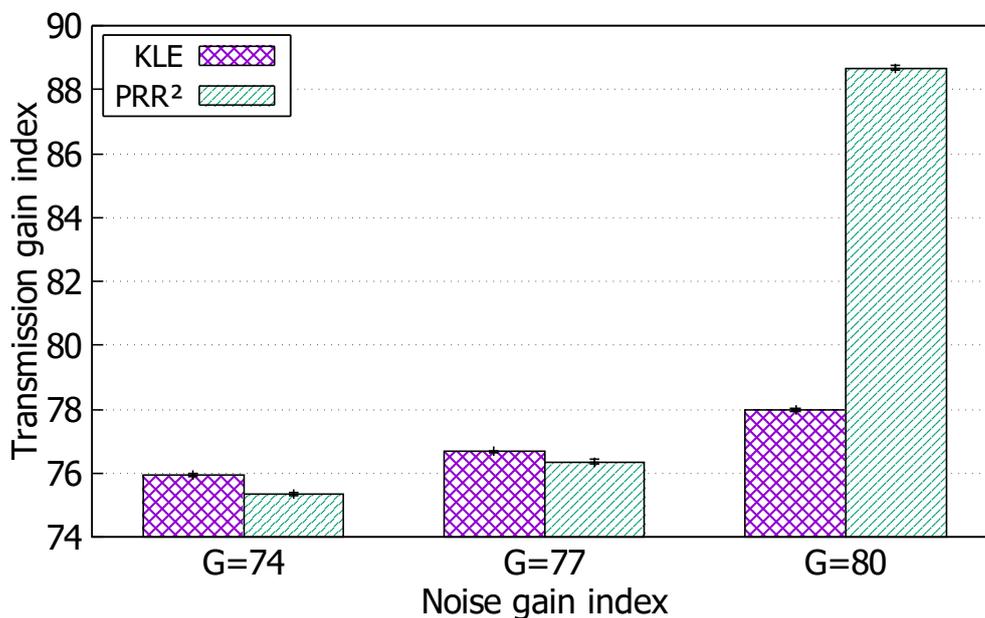


Figura 6.12. Ganho na transmissão no segundo cenário

6.2.5 Discussão

Com os dois cenários avaliados foi possível validar a abordagem do uso de informações provenientes de duas camadas de rede para produzir uma estimativa precisa em

ambientes ruidosos. Com o primeiro cenário foi ratificada a estratégia de se utilizar duas janelas de PRR, de curto e longo alcances. No segundo cenário foi possível verificar o desempenho do estimador em comparação com o estimador KLE. O PRR² proporcionou maiores taxas de entrega em comparação com o KLE, com uma taxa de retransmissões inferior, e com um menor incremento na potência de transmissão. Os resultados levaram a conclusões similares nos cenários em que se variou o intervalo de transmissão entre as mensagens e naqueles que se variou a potência de transmissão do ruído artificial.

Em algumas situações, como o apresentado na Figura 6.12 ($G = 80$ dB), nota-se a importância do uso de uma métrica baseada na contagem de pacotes, como PRR: como o gerador de fonte de ruído está situado mais próximo do USRP receptor do que do USRP transmissor (Figura 6.3), o RSSI obtido pelo remetente não é capaz de capturar o ruído na área do receptor. Observa-se que o KLE não foi suficiente para alcançar boas taxas de entrega nesses casos (elevados ruídos), apresentando grande variabilidade nos resultados, enquanto o PRR², utilizando uma estratégia simples que requer pequeno poder de computação, teve um comportamento mais estável. Dessa forma, os intervalos de confiança mais baixos apresentados no PRR² podem ser interpretados como uma maior estabilidade nas estimativas.

6.3 Resultados dos experimentos com LQM

Esta seção apresenta a avaliação experimental do estimador proposto LQM utilizando *hardware* programável. Inicialmente é descrita a implementação do PRR², seguida da descrição dos cenários experimentais, e por fim apresentados os resultados obtidos.

6.3.1 Implementação do LQM no USRP

O estimador LQM é avaliado em um protótipo utilizando rádios definidos por *software*. Criamos um experimento que representa a transmissão de dados entre dois dispositivos de baixa potência, por exemplo, um monitor de frequência cardíaca enviando pacotes para um coletor. Nesse cenário, o objetivo principal é transmitir dados de maneira confiável, consumindo a menor quantidade de energia. Para esse fim, usamos estimativas de qualidade do enlace para ajustar dinamicamente a transmissão de energia.

A disposição dos equipamentos USRPs está ilustrada na Figura 6.13. Foram utilizados cinco Ettus USRPs executando o protocolo IEEE 802.15.4 como nós

de comunicação: quatro transmissores e um receptor. Outras três USRPs (cores vermelhas) podem ser utilizadas para gerar ruído gaussiano na mesma frequência operacional das outras USRPs.

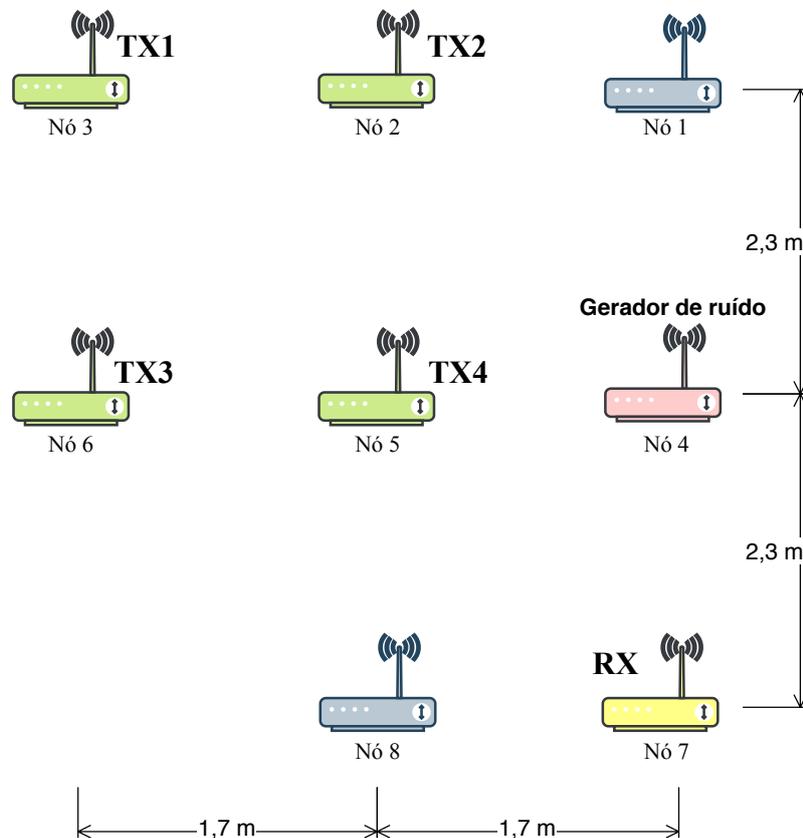


Figura 6.13. Disposição dos USRPs para o experimento de avaliação do LQM

Os USRPs executam módulos do *software* GNU Radio. Nós implementamos os módulos para obter as métricas de entrada (RSSI e SNR) e calcular o LQE, além de modificações na pilha de protocolos. Empregamos uma versão aprimorada do protocolo IEEE 802.15.4 originalmente desenvolvido por Bloessl et al. [2013]. A implementação original não fornece uma camada MAC funcional, ou seja, não há detecção de transportadora e nem retransmissões. Foram implementadas a detecção de portadora e o algoritmo de retorno exponencial binário para retransmissões [Cordeiro et al., 2018]². A potência de transmissão pode ser alterada dinamicamente e, por esta razão, é um valor calculado em um módulo separado que implementa o LQE. Todo o código implementado, incluindo LQM e as modificações nas camadas

²<https://github.com/jeffRayneres/FS-MAC>

física e MAC do GNU Radio, estão disponíveis publicamente em um repositório GIT³.

O índice de ganho de transmissão (TGI - *Transmission gain index*) é o valor em dBm usado pelo atenuador no USRP. Na USRP B210, o índice de ganho varia de 0 a 89 dB, e no modelo B100, o índice de ganho varia de 0 a 35 dB.

Os resultados mostrados nesta seção são uma média de cinco execuções independentes de cada configuração. Devido aos curtos intervalos de confiança obtidos nos experimentos, cinco execuções indicaram ser suficientes para obter resultados conclusivos. Cada execução foi realizada com a transmissão de 250 pacotes de 100 bytes, além das retransmissões. Os pacotes são gerados periodicamente a cada 400 ms, e os resultados são plotados com um intervalo de confiança de 95%.

Baselines: o estimador LQM é comparado com dois protocolos da literatura - *wmewmaPRR* [Woo & Culler, 2003] e *ETX* [Couto et al., 2003]. O estimador proposto também é comparado para um caso hipotético (chamado *MaxGain*), no qual as transmissões são sempre realizadas com a potência máxima de transmissão.

Nas próximas subseções, os resultados do experimento serão apresentados em três cenários: no primeiro (C1), existe apenas um remetente e um receptor, sem a adição de ruído artificial. No segundo cenário (C2), há um remetente e um receptor na presença de ruído artificial e, no terceiro cenário (C3), existem múltiplos remetentes e um receptor, sem ruído artificial. Esses cenários estão resumidos na Tabela 6.2.

Tabela 6.2. Descrição dos cenários do experimento com LQM

Cenário	Descrição
C1	Um transmissor, sem ruído artificial
C2	Um transmissor, com ruído artificial
C3	Múltiplos transmissores; sem ruído artificial

6.3.2 Resultados

6.3.2.1 Primeiro cenário: comunicação sem adição de ruído artificial

Este cenário compara o estimador LQM com o *ETX*, *wmewmaPRR* (chamado apenas *PRR*) e *MaxGain*, bem como com o LQM sem a detecção de *Concept Drift* (*CD*), todos descritos anteriormente, considerando apenas um transmissor e um receptor, sem a adição de ruído artificial. Este cenário visa investigar as comunicações

³<https://github.com/wendley/gr-lqe>

entre dois dispositivos em condições ideais, o que no mundo real significa pouco ou nenhum ruído artificial na transmissão.

Na Figura 6.14, exceto para o MaxGain, o algoritmo que alcançou o melhor PDR foi o LQM sem CD. O LQM melhorou a taxa de entrega em 16,9% em comparação com o PRR e reduziu o TPRP em 54,6%, em média, como mostrado no lado direito da mesma figura. O LQM com CD foi o segundo melhor estimador, com uma discreta diferença de 0,67 pontos percentuais em relação ao LQM sem CD. Quando comparado ao ETX, o LQM sem CD, melhorou a taxa de entrega em 1,6%, com uma redução de 27,8% nas retransmissões de pacotes. Um número menor de retransmissões também é importante para uma rede confiável, pois reduz os atrasos na entrega de pacotes. Além disso, estimadores precisos também reduziram os intervalos de confiança, reduzindo a instabilidade, o que é importante para aplicações de controle em tempo real.

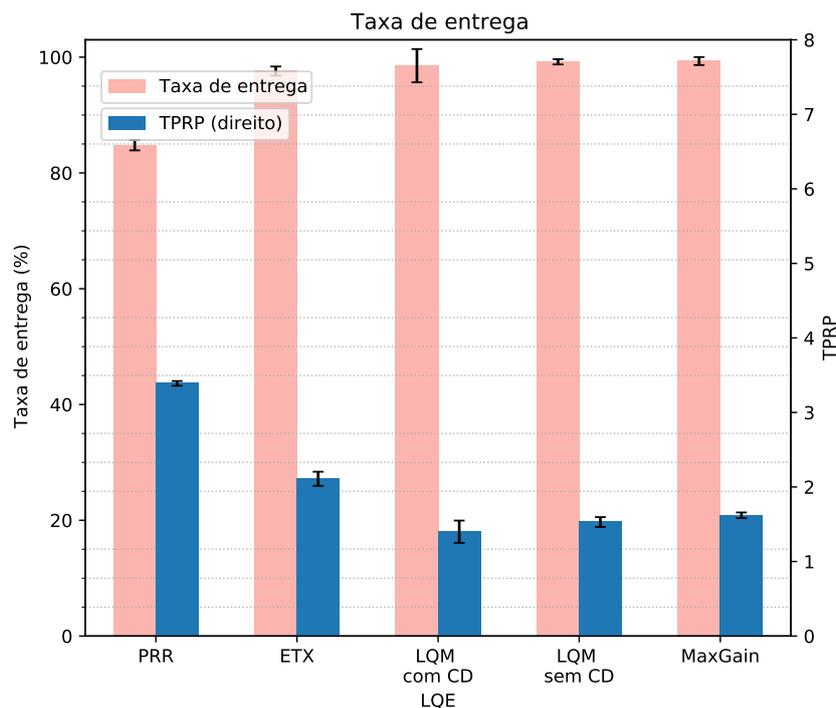


Figura 6.14. Taxa de entrega e TPRP sem adição de ruído artificial

A Figura 6.15 apresenta o total de energia utilizada em cada transmissão multiplicada pelo número de transmissões. Quando analisamos este parâmetro, o melhor resultado é indicado por um valor mais baixo. O estimador LQM sem CD consumiu menos de 24,7% quando comparado ao PRR. O estimador hipotético

Max-Gain (transmitindo com a potência máxima) consumiu 27,7% a mais do que o LQM.

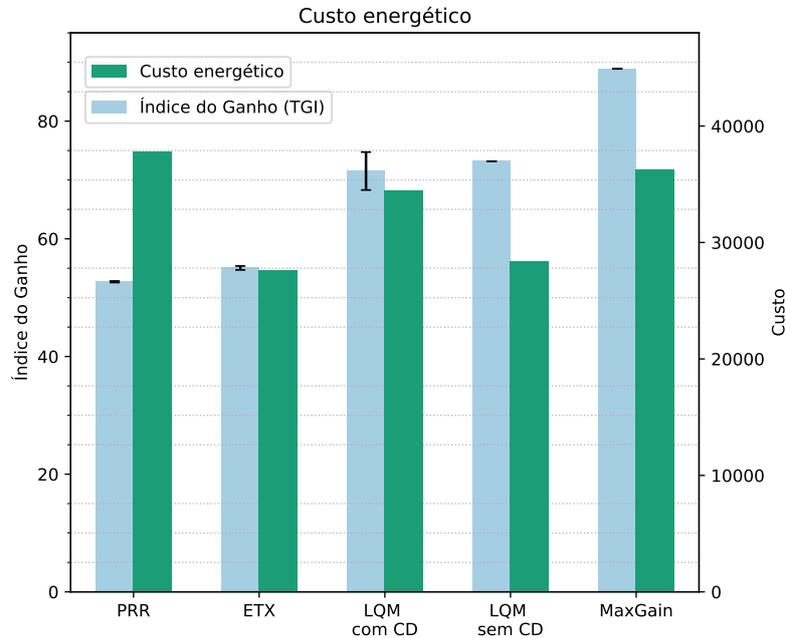


Figura 6.15. TGI e custo energético sem adição de ruído artificial

6.3.2.2 Segundo cenário: comunicação com adição de ruído artificial

Este cenário introduz o ruído artificial no cenário experimental e compara o LQM com o ETX, PRR e MaxGain, bem como com o LQM com detecção de *Concept Drift* (CD), considerando apenas um transmissor e um receptor. Como observado na Figura 6.16, excetuando o hipotético estimador MaxGain, o algoritmo que atingiu a melhor taxa de entrega foi LQM sem CD. As técnicas de aprendizagem de máquina propiciaram ao LQM melhorar a taxa de entrega em 43,8% em comparação com o PRR e reduziu o TPRP em 61,6%, em média, como mostrado no lado direito da mesma figura. O LQM com detecção de CD foi o segundo melhor estimador, com uma diferença de 11,7% em relação ao LQM sem CD.

Quando comparado ao estimador ETX, o LQM sem CD melhorou a taxa de entrega em 20,3%, com uma redução de 47,7% nas retransmissões de pacotes. O custo de energia, apresentado na Figura 6.17, indica que o LQM obteve o melhor resultado (valor mais baixo), com uma melhoria de 35,0% quando comparado ao

PRR e 39,9% melhor que o ETX. O MaxGain consumiu 29,9% mais energia que o LQM.

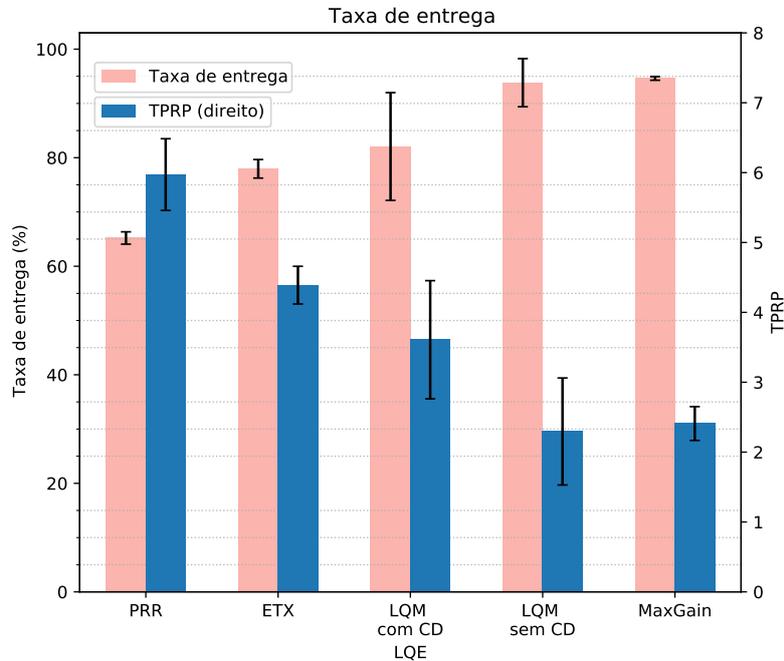


Figura 6.16. Taxa de entrega e TPRP com adição de ruído artificial

6.3.2.3 Terceiro cenário: comunicação com múltiplos remetentes

Este cenário envolve quatro equipamentos transmissores e um receptor, como mostra a Figura 6.13. Conforme observado na Figura 6.18, o LQM sem CD foi o melhor estimador em 3 dos 4 nós. Naquele nó em que o LQM não foi o melhor (nó 16), o melhor resultado foi alcançado com o ETX. As figuras 6.19 e 6.20 apresentam a taxa de entrega média e o consumo de energia para os nós. O LQM sem detecção de CD alcançou uma taxa de entrega mais alta (99,3%) quando comparado ao PRR (91,1%) e ETX (98,1%). Ele também obteve a melhor taxa de pacotes entregues de todos os estimadores, atrás apenas do hipotético MaxGain, como mostra a Figura 6.19. O melhor resultado em termos de custo de energia foi obtido pelo ETX, com o LQM sem CD demandando 11,2% mais energia.

Apesar do indicativo teórico de melhores resultados na adoção de *Machine Learning* com detecção de *concept drift*, o experimento prático mostrou pequenas diferenças com e sem detecção de CD. Havia uma pequena vantagem para o ML sem

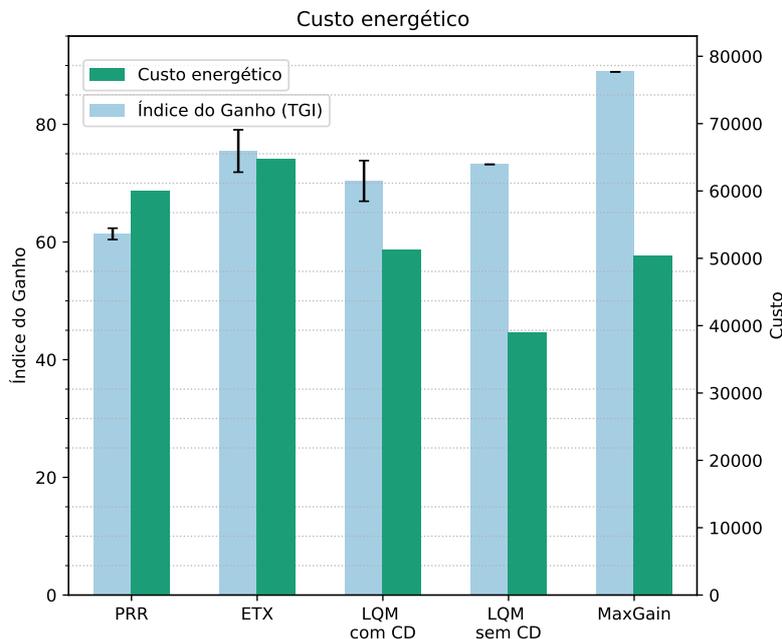


Figura 6.17. TGI e custo de energia com adição de ruído artificial

a detecção de CD, e isso pode indicar que o tempo de treinamento de um novo banco de dados e seu tamanho pequeno podem comprometer o resultado em cenários com muitas transmissões de pacotes. Novas experiências em trabalhos futuros fornecerão mais indicativos para definir se o tratamento de CD é útil nesse contexto.

6.3.3 Discussão

O estimador LQM foi avaliado em três configurações e, em todas elas, o LQM foi comparado com os estimadores `wmewmaPRR` e o ETX, além do hipotético MaxGain. No primeiro cenário, executamos o estimador com um único nó enviando mensagens. No segundo cenário, um ruído artificial gaussiano foi adicionado ao experimento usando o USRP. Em ambos, o LQM supera a taxa de entrega do PRR e ETX, com menos retransmissões e com um discreto aumento na potência de transmissão. O terceiro cenário considerou quatro transmissores e um receptor.

Algumas situações, como a apresentada na Figura 6.16, destacam a importância de uma métrica baseada em várias entradas, como o LQM: como a fonte de ruído está situada mais perto do receptor USRP do que do remetente, o RSSI obtido pelo remetente não pondera de forma adequada o ruído próximo ao lado do receptor. À

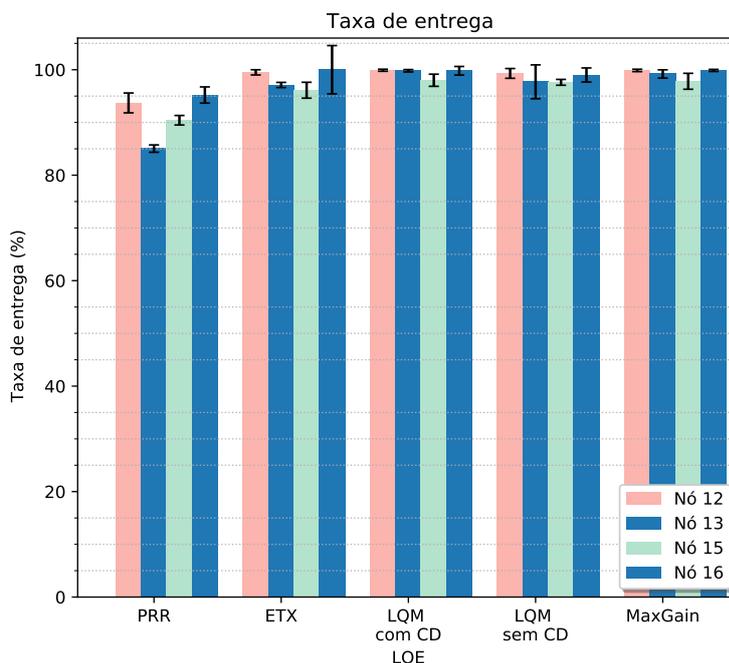


Figura 6.18. Taxa de entrega - cenário com múltiplos remetentes

medida que o estimador incorpora mais métricas de decisão, ele tende a fornecer uma estimativa mais acurada.

Além disso, observa-se que o PRR tradicional não foi suficiente para obter boas taxas de entrega no cenário com ruído artificial, enquanto o LQM e o ETX obtiveram melhor desempenho.

Os resultados também sugerem que o tratamento para o *concept drift* pode não propiciar melhorias imediatas no sistema, o que pode ser explicado pelas características do CD. Segundo Gama et al. [2014], a detecção de CD permite identificar duas implicações dos desvios: (i) se a distribuição de dados muda e afeta a decisão preditiva e (ii) se as mudanças são visíveis a partir da distribuição de dados sem conhecer os verdadeiros rótulos. De uma perspectiva preditiva, apenas as mudanças que afetam a decisão de previsão requerem adaptação ou tratamento. Além disso, embora o processo de adaptação vise melhorar o resultado preditivo, essa melhoria depende da natureza dos dados, que mesmo com o tratamento de *concept drift* podem não fornecer melhores resultados em alguns tipos de dados.

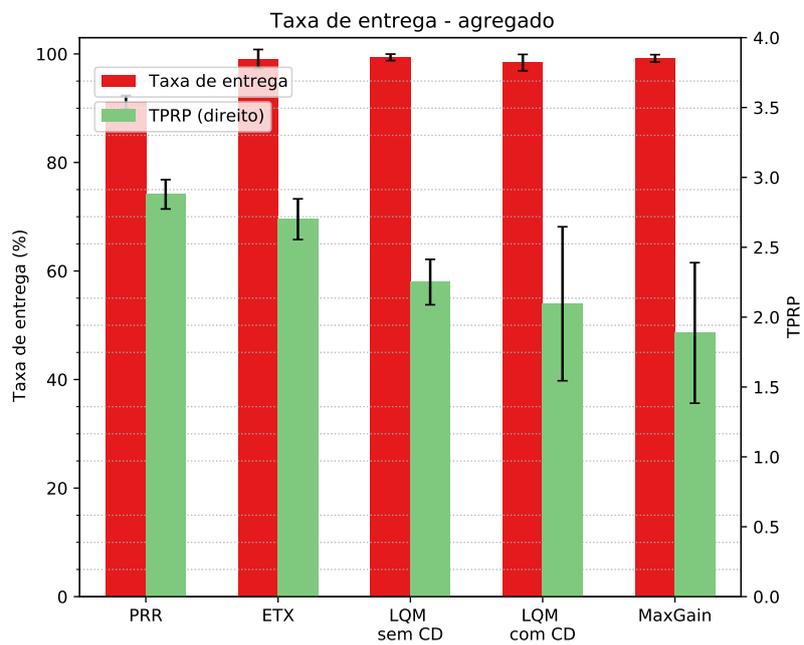


Figura 6.19. Taxa de entrega e TPRP - cenário de múltiplos remetentes agregados

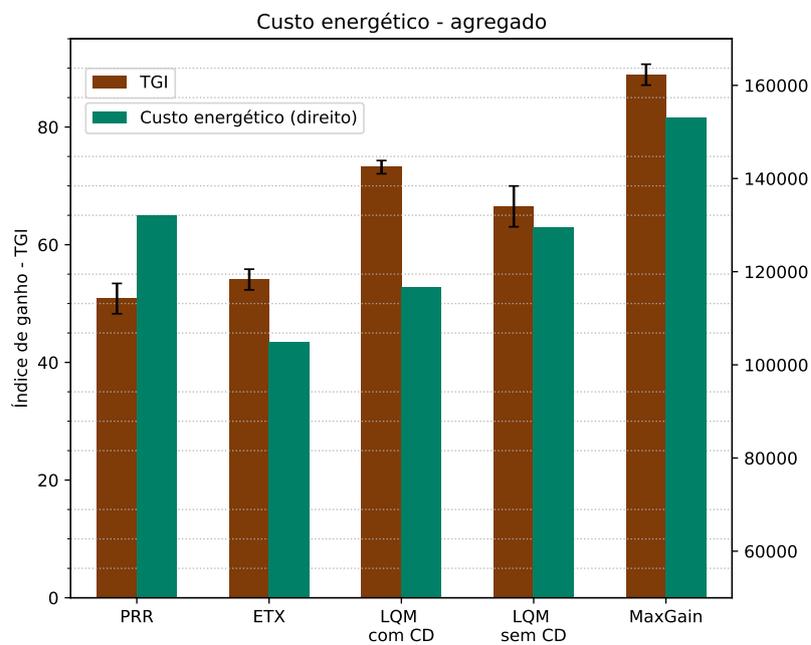


Figura 6.20. TGI e custo energético - cenário de múltiplos remetentes agregado

Capítulo 7

Considerações finais

Este capítulo apresenta as conclusões deste trabalho, as limitações relacionadas aos experimentos e, por fim, apresenta as possibilidades de trabalhos futuros.

7.1 Conclusões

Os estimadores de qualidade de enlace precisos são ferramentas fundamentais para o funcionamento eficiente de diversos protocolos de rede. A estimativa produzida geralmente é obtida pela avaliação de uma ou mais métricas de enlace dentro de uma janela de estimação. Este trabalho discutiu as principais técnicas para estimação de qualidade do enlace. Foram apresentadas as subdivisões dos LQEs baseando-os em métricas físicas ou em métricas lógicas. Os LQEs também podem ser classificados baseando-os em potência do sinal ou em estatísticas de pacote, sendo esta a classificação adotada neste trabalho.

O principal objetivo deste trabalho foi o estudo de métodos e técnicas de estimação de qualidade de enlace em comunicação M2M, bem como a proposição de novos estimadores para melhoria da confiabilidade na transmissão entre dispositivos M2M. Adicionalmente, este trabalho objetivou avaliar o efeito do processo de estimação em sistemas de comunicação M2M em condições adversas, como na presença de ruído eletromagnético. Para alcançar os objetivos esperados, foram desenvolvidos dois estimadores multicamadas denominados PRR² e LQM, este último com a aplicação de técnicas de *machine learning*, com a detecção de *concept drift*.

O PRR² recebe como entradas três parâmetros: o valor de RSSI, o pacote *ack*, e o aviso de *timeout* de espera do *ack*. A saída do PRR² é a estimativa produzida da qualidade do enlace. Um diferencial do PRR² é que ele utiliza dois PRRs, provenientes de duas janelas de observação: uma de curta e outra de longo alcance.

Os experimentos práticos para validar o desempenho do PRR² utilizaram os *hardwares* programáveis USRP modelo B210. Dois cenários experimentais foram montados: no *primeiro cenário* foi verificado o efeito de cada um dos fatores inerentes ao PRR², como o RSSI, o PRR, o RSSI+PRR, o PRR com duas janelas de observação, e o RSSI+PRR com duas janelas de observação.

Os resultados mostraram que a principal contribuição do PRR² advém da combinação do RSSI com o PRR de duas janelas de observação, o que permite capturar de forma mais estável variações da qualidade do enlace no curto prazo, enquanto mantém informações da qualidade de longo prazo. Também foi verificado que o PRR² contribui para a diminuição das retransmissões de pacotes, o que pode implicar em melhor eficiência energética.

No *segundo cenário* experimental do PRR², o objetivo foi comparar o desempenho do PRR² com o KLE. Os resultados indicaram que o PRR² provê uma melhor taxa de entrega dos pacotes, à medida que identifica de forma mais rápida e eficiente a qualidade do enlace, sem demandar aumentos significativos da potência de transmissão. O resultado supramencionado foi alcançado com uma redução significativa da quantidade de retransmissões de pacotes, em comparação com o estimador KLE.

Os resultados sugerem que o processo de estimação da qualidade do enlace rápido e preciso em sistemas de comunicação M2M em condições adversas minimizam os efeitos negativos da degradação da qualidade do enlace. No nosso experimento, isso se dá principalmente em razão da estimativa precisa fornecer os indicadores adequados para ajustar os parâmetros de potência de transmissão em valores suficientes a superar as dificuldades impostas pela degradação do canal.

O uso de técnicas de aprendizagem de máquina para a produção de uma estimativa de qualidade do enlace foi avaliado com o uso de oito algoritmos e utilizou uma abordagem que envolve um pré-processamento dos dados e detecção e adaptação de *concept drift* (CD).

O estimador LQM proposto utiliza *machine learning* com detecção de CD para fornecer uma estimativa mais acurada da qualidade do enlace. O LQM utiliza os valores de RSSI, SNR, PRR, latência e TPRP para computar a sua estimativa, e adotou o algoritmo *Decision Tree Regressor* para a avaliação. Os resultados de avaliação do estimador LQM indicam que ele é capaz de fornecer uma melhor taxa de entrega de pacotes, ao mesmo tempo em que possibilita diminuir o número de retransmissões quando comparado com os estimadores PRR e ETX da literatura. Além disso, o LQE proposto é mais estável que os outros estimadores avaliados, com apenas um discreto aumento no consumo energético nos equipamentos de transmissão.

Os resultados também sugerem que uma vez detectado *concept drift* a adaptação do modelo de aprendizado adotada neste trabalho pode não propiciar melhorias imediatas no sistema, o que pode ser explicado pelas características inerentes a um CD e pelo tempo necessário para retreino. Ademais, embora a adaptação geralmente melhore os resultados do modelo, essa melhoria é fortemente dependente da natureza dos dados.

7.2 Limitações

Há diversas limitações inerentes a um experimento prático, o que delimita o escopo de análise da pesquisa. Este trabalho foca principalmente nos aspectos de rede do estimador de qualidade do enlace, ou seja, seu desempenho no mundo real em redes sem fio. Portanto, evitamos abordar em detalhes cada algoritmo de aprendizado de máquina. Os leitores interessados podem aprender mais sobre cada algoritmo nas referências fornecidas neste trabalho.

O uso do *testbed* FUTEBOL/UFMG possibilitou a execução de diversos experimentos práticos. Entretanto, pela sua instalação fixa, impossibilita, ou dificulta, a realização de um conjunto de testes considerando cenários com eventual mobilidade.

A segurança e a privacidade dos dados não foram avaliados neste trabalho, principalmente porque demandaria um esforço computacional adicional, o que influenciaria na execução dos algoritmos de aprendizagem de máquina. Ademais, promovemos a transmissão de dados tratando-os principalmente nas camadas física e enlace de dados. Segurança e privacidade, de modo geral, podem ser melhor trabalhados utilizando mecanismos que atuem em camadas superiores, como a camada de aplicação.

7.3 Trabalhos futuros

As atividades de pesquisa que podem ser desenvolvidas a partir desse trabalho consideram as seguintes ideias:

Novos cenários: esta ideia sugere desenvolver os experimentos com os estimadores de qualidade do enlace em novos cenários, tais como cenários com intensidade de ruído artificial variável, objetivando emular ambientes industriais, onde é comum a presença de ruído com intensidade irregular, e/ou considerar um cenário com mobilidade;

Incorporação de um estimador em um protocolo de comunicação sem fio:

esta tarefa facilitaria a coleta de mais informações sobre a qualidade do enlace no lado do receptor. Para isso, poderá ser configurado no lado receptor um módulo de coleta e cálculo do RSSI e o seu valor médio será esporadicamente enviado ao equipamento transmissor;

Utilização de outras plataformas:

esta ideia visa explorar outros *hardwares* para execução dos algoritmos estimadores de qualidade, tais como os *single-board*, como Raspberry, ou placas de prototipagem eletrônica de *hardware* livre, como o Arduino e NodeMCU. Isso permitiria a inovação com o desenvolvimento de novas técnicas e algoritmos nesses equipamentos que apresentam severas limitações de recursos;

Estimadores de qualidade do enlace independente de resposta (*ack*):

a maioria dos estimadores de múltiplas camadas utilizam a confirmação do pacote recebido (*ack*) como um indicador para atualizar suas métricas de estimativa, entretanto, alguns transmissores em situações específicas podem não receber o *ack*. O desafio neste caso é desenvolver um estimador que atualize suas métricas mesmo na ausência, ou longo atraso, de um *ack*, o que dificultaria até mesmo o cálculo do estimador PRR;

Calibração automática do RSSI:

de acordo com o *hardware* utilizado, os valores de RSSI podem ter várias graduações para um mesmo nível de potência de sinal residual ou de transmissão. Uma funcionalidade importante que ajudaria a escalabilidade de um LQE seria a implantação de um sistema de calibração automática dos valores mínimos e máximos de RSSI para aquele equipamento em uso.

7.4 Agradecimentos

Este trabalho foi parcialmente financiado pelas agências MCTI/CGI-RNP e Fapesp (EU-BR-Futebol); CNPQ-Universal 2016 (Projeto Unicom) e Fapemig-Universal 2016 (Projeto Unidad).

Referências Bibliográficas

- Akkaya, K. & Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325--349.
- Alexiou, A. & Gotsis, A. (2015). 10 - packet scheduling strategies for machine-to-machine (m2m) communications over long-term evolution (lte) cellular networks. Em Antón-Haro, C. & Dohler, M., editores, *Machine-to-machine (M2M) Communications*, pp. 173 – 186. Woodhead Publishing, Oxford.
- Anton-Haro, C. & Dohler, M. (2014). *Machine-to-machine (M2M) Communications: Architecture, Performance and Applications*. Elsevier.
- Asadi, A.; Jacko, P. & Mancuso, V. (2014). Modeling d2d communications with lte and wifi. *ACM SIGMETRICS Performance Evaluation Review*, 42(2):55--57.
- Åström, K. J. & Murray, R. M. (2010). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.
- Axell, E.; Leus, G.; Larsson, E. G. & Poor, H. V. (2012). Spectrum sensing for cognitive radio: State-of-the-art and recent advances. *Signal Processing Magazine, IEEE*, 29(3):101--116.
- Baccour, N.; Koubaa, A.; Jamâa, M. B.; Do Rosario, D.; Youssef, H.; Alves, M. & Becker, L. B. (2011). Radiale: A framework for designing and assessing link quality estimators in wireless sensor networks. *Ad Hoc Networks*, 9(7):1165--1185.
- Baccour, N.; Koubaa, A.; Mottola, L.; Zuniga, M. A.; Youssef, H.; Boano, C. A. & Alves, M. (2012). Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34.
- Baccour, N.; Koubâa, A.; Youssef, H. & Alves, M. (2015). Reliable link quality estimation in low-power wireless networks and its impact on tree-routing. *Ad Hoc Networks*, 27:1--25. ISSN 15708705.

- Baccour, N.; Koubâa, A.; Youssef, H.; Jamâa, M. B.; Do Rosario, D.; Alves, M. & Becker, L. B. (2010). F-lqe: A fuzzy link quality estimator for wireless sensor networks. Em *Wireless Sensor Networks*, pp. 240--255. Springer.
- Baheti, R. & Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12:161--166.
- Basseville, M.; Nikiforov, I. V. et al. (1993). *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs.
- Benesty, J.; Chen, J.; Huang, Y. & Cohen, I. (2009). Pearson correlation coefficient. Em *Noise reduction in speech processing*, pp. 1--4. Springer.
- Bhat, P. & Dohler, M. (2015). Overview of 3gpp machine-type communication standardization. Em *Machine-to-machine (M2M) Communications*, pp. 47--62. Elsevier.
- Biaz, S. & Wu, S. (2008). Rate adaptation algorithms for iee 802.11 networks: A survey and comparison. Em *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pp. 130--136. IEEE.
- Bifet, A. & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. Em *Proceedings of the 2007 SIAM international conference on data mining*, pp. 443--448. SIAM.
- Bifet, A. & Gavaldà, R. (2009). Adaptive learning from evolving data streams. Em *International Symposium on Intelligent Data Analysis*, pp. 249--260. Springer.
- Bifet, A.; Holmes, G.; Pfahringer, B.; Kirkby, R. & Gavaldà, R. (2009). New ensemble methods for evolving data streams. Em *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 139--148. ACM.
- Bloessl, B.; Leitner, C.; Dressler, F. & Sommer, C. (2013). A GNU Radio-based IEEE 802.15. 4 Testbed. *12. GI/ITG FACHGESPRÄCH SENSORNETZE*, p. 37.
- Boano, C. A.; Zúniga, M. A.; Voigt, T.; Willig, A. & Romer, K. (2010). The triangle metric: Fast link quality estimation for mobile wireless sensor networks. Em *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pp. 1--7. IEEE.

- Burns, A.; Greene, B. R.; McGrath, M. J.; O'Shea, T. J.; Kuris, B.; Ayer, S. M.; Stroiescu, F. & Cionca, V. (2010). ShimmerTM—a wireless sensor platform for noninvasive biomedical research. *Sensors Journal, IEEE*, 10(9):1527--1534.
- Buskirk, T. D. (2018). *An Introduction to Machine Learning Methods for Survey Researchers*. Tese de doutorado, University of Rochester.
- Cacciapuoti, A. S.; Caleffi, M.; Paura, L. & Rahman, M. A. (2014). Link quality estimators for multi-hop mesh network. Em *Euro Med Telco Conference (EMTC), 2014*, pp. 1--6. IEEE.
- Caleffi, M. & Paura, L. (2009). Bio-inspired link quality estimation for wireless mesh networks. Em *World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*, pp. 1--6. IEEE.
- Cármenes, R. S. (1991). Lstsq: a module for reliable constrained and unconstrained nonlinear regression. *Bioinformatics*, 7(3):373--378.
- Chang, C.-C. & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1--27.
- Chen, M.; Wan, J.; Gonzalez, S.; Liao, X. & Leung, V. C. M. (2014). A survey of recent developments in home M2M networks. *IEEE Communications Surveys and Tutorials*, 16(1):98--114. ISSN 1553877X.
- Chen, S.; Ma, R.; Chen, H.-H.; Zhang, H.; Meng, W. & Liu, J. (2017). Machine-to-machine communications in ultra-dense networks—a survey. *IEEE Communications Surveys & Tutorials*, 19(3):1478--1503.
- Cisco (2019). Global mobile data traffic forecast update, 2017-2022. http://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/index.html.
- Cisco (2020). Cisco annual internet report, 2018-2023. <https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/air-highlights.html>.
- Clare, L. P.; Pottie, G. J. & Agre, J. R. (1999). Self-organizing distributed sensor networks. Em *AeroSense'99*, pp. 229--237. International Society for Optics and Photonics.
- Cordeiro, J. R.; Macedo, D. F. & Vieira, L. F. (2018). Fs-mac: A flexible mac platform for wireless networks. Em *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1--6. IEEE.

- Cotuk, H.; Bicakci, K.; Tavli, B. & Uzun, E. (2014). The impact of transmission power control strategies on lifetime of wireless sensor networks. *Computers, IEEE Transactions on*, 63(11):2866--2879.
- Couto, D. S.; Aguayo, D.; Morris, R. & Bicket, J. (2003). A high-throughput path metric for multi-hop wireless routing.
- da Silva, W. S.; Macedo, D. F.; Nogueira, M.; Nguyen, T. M. T. & Nogueira, J. M. S. (2017). A multilayer link quality estimator for reliable machine-to-machine communication. Em *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1--7. IEEE.
- De Couto, D. S.; Aguayo, D.; Bicket, J. & Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wireless networks*, 11(4):419--434.
- Demetri, S.; Zúñiga, M.; Picco, G. P.; Kuipers, F.; Bruzzone, L. & Telkamp, T. (2019). Automated estimation of link quality for lora: a remote sensing approach. Em *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 145--156. IEEE.
- Di Caro, G. A.; Kudelski, M.; Flushing, E. F.; Nagi, J.; Ahmed, I. & Gambardella, L. M. (2013). Online supervised incremental learning of link quality estimates in wireless networks. Em *2013 12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pp. 133--140. IEEE.
- Doddavenkatappa, M.; Chan, M. C. & Ananda, A. L. (2011). Indriya: A low-cost, 3d wireless sensor network testbed. Em *International conference on testbeds and research infrastructures*, pp. 302--316. Springer.
- Dumont, M.; Marée, R.; Wehenkel, L. & Geurts, P. (2009). Fast multi-class image annotation with random subwindows and multiple output randomized trees. Em *Proc. International Conference on Computer Vision Theory and Applications (VI-SAPP)*, volume 2, pp. 196--203.
- Dutra Souza, A.; Marques, A. F.; Macedo, D. F.; Collins, D.; Júnior, G. M.; Cordeiro, J. R.; Marquez-Barja, J. M.; Kayo Coelho, K.; Pinto, L. R.; da Silva, L. A. et al. (2017). Experimental wireless networking research using software-defined radios. Em *Brazilian Symposium on Computer Networks and Distributed Systems-XXXV SBRC 2017*, pp. 1--49.

- Engstrom, J. C. & Gray, C. (2008). Clear channel assessment in wireless sensor networks. Em *Proceedings of the 46th Annual Southeast Regional Conference on XX*, pp. 464--468.
- Ferng, H.-W. & Abdullah, M. (2019). Mobility-based clustering with link quality estimation for urban vanets. Em *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 1--7. IEEE.
- Fonseca, R.; Gnawali, O.; Jamieson, K. & Levis, P. (2007). Four-bit wireless link estimation. Em *HotNets*.
- Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M. & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):44.
- Gao, W.; Jiao, B.; Yang, G.; Hu, W. & Liu, J. (2014). Transmission power control for IEEE 802.15. 6 body area networks. *ETRI Journal*, 36(2):313--316.
- Gomes, E. B.; Gomes, R. D. & Fonseca, I. E. (2019). Evaluation of a link quality estimator in an outdoor wsn using a dedicated node. Em *2019 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1204--1209. IEEE.
- Gomes, H. M.; Bifet, A.; Read, J.; Barddal, J. P.; Enembreck, F.; Pfharinger, B.; Holmes, G. & Abdessalem, T. (2017a). Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9-10):1469--1495.
- Gomes, R. D.; Queiroz, D. V.; Lima Filho, A. C.; Fonseca, I. E. & Alencar, M. S. (2017b). Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes. *Ad Hoc Networks*, 59:116--133.
- Gubbi, J.; Buyya, R.; Marusic, S. & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645--1660.
- Guo, G.; Wang, H.; Bell, D.; Bi, Y. & Greer, K. (2003). Knn model-based approach in classification. Em *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pp. 986--996. Springer.
- Heinzer, P. (2011). *Wireless Link Quality Estimation in Mobile Networks*. Master thesis, Swiss Federal Institute of Technology Zurich.

- Hellerstein, J. L.; Diao, Y.; Parekh, S. & Tilbury, D. M. (2004). *Feedback control of computing systems*. John Wiley & Sons.
- Holler, J.; Tsiatsis, V.; Mulligan, C.; Avesand, S.; Karnouskos, S. & Boyle, D. (2014). *From machine-to-machine to the internet of things - Introduction to a New Age of Intelligence*. Academic Press, Elsevier.
- Huang, Z.; Yee, L.; Ang, T. F.; Anisi, M. H. & Adam, M. S. (2019). Improving the accuracy rate of link quality estimation using fuzzy logic in mobile wireless sensor network. *Advances in Fuzzy Systems*, 2019.
- Hussain, F.; Anpalagan, A. & Vannithamby, R. (2017). Medium access control techniques in m2m communication: survey and critical review. *Transactions on Emerging Telecommunications Technologies*, 28(1):e2869.
- Hyndman, R. J. & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- IEEE (2013). IEEE 802.15.4 Standard. <https://standards.ieee.org/getieee802/download/802.15.4k-2013.pdf>. Viewed in Jun 2016.
- Jones, T. (2017). Models for machine learning. <https://developer.ibm.com/articles/cc-models-machine-learning>. Viewed in abr 2019.
- Kartsakli, E.; Lalos, A. S.; Antonopoulos, A.; Tennina, S.; Renzo, M. D.; Alonso, L. & Verikoukis, C. (2014). A survey on M2M systems for mHealth: a wireless communications perspective. *Sensors*, 14(10):18009--18052.
- Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q. & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Em *Advances in neural information processing systems*, pp. 3146--3154.
- Kim, J.; Lee, J.; Kim, J. & Yun, J. (2014). M2m service platforms: Survey, issues, and enabling technologies. *IEEE Communications Surveys & Tutorials*, 16(1):61-76.
- King, A. & Roedig, U. (2018). Differentiating clear channel assessment using transmit power variation. *ACM Transactions on Sensor Networks (TOSN)*, 14(2):1--28.
- LaI, D.; Manjeshwar, A.; Herrmann, F.; Uysal-Biyikoglu, E. & Keshavarzian, A. (2003). Measurement and characterization of link quality metrics in energy constrained wireless sensor networks. Em *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, volume 1, pp. 446--452. IEEE.

- Lee, W.-M. (2019). *Python Machine Learning*. John Wiley & Sons.
- Lien, S.-Y.; Chen, K.-C. & Lin, Y. (2011). Toward ubiquitous massive accesses in 3gpp machine-to-machine communications. *IEEE Communications Magazine*, 49(4).
- Lin, S.; Miao, F.; Zhang, J.; Zhou, G.; Gu, L.; He, T.; Stankovic, J. A.; Son, S. & Pappas, G. J. (2016). Atpc: adaptive transmission power control for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 12(1):1--31.
- Lin, Y.-D. & Hsu, Y.-C. (2000). Multihop cellular: A new architecture for wireless communications. Em *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pp. 1273--1282. IEEE.
- Liu, T. & Cerpa, A. E. (2014). Data-driven link quality prediction using link features. *ACM Transactions on Sensor Networks*, 10(2):1--35. ISSN 15504859.
- Lowrance, C. J. & Lauf, A. P. (2017). Link quality estimation in ad hoc and mesh networks: A survey and future directions. *Wireless Personal Communications*, 96(1):475--508.
- Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J. & Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*.
- Luo, H.; Huang, Z. & Zhu, T. (2015). A survey on spectrum utilization in wireless sensor networks. *Journal of Sensors*, 2015.
- Makridakis, S.; Wheelwright, S. C. & Hyndman, R. J. (2008). *Forecasting methods and applications*. John Wiley & Sons.
- Manly, B. F. & Mackenzie, D. (2000). A cumulative sum type of method for environmental monitoring. *Environmetrics: The official journal of the International Environmetrics Society*, 11(2):151--166.
- Misic, V. & Misic, J. (2014). *Machine-to-Machine Communications: Architectures, Technology, Standards, and Applications*. Taylor & Francis. ISBN 9781466561236.
- Morettin, P. A. & de Castro Toloí, C. M. (1981). *Modelos para previsão de séries temporais*, volume 1. Instituto de matematica pura e aplicada.
- Morettin, P. A. & Toloí, C. (2006). *Análise de séries temporais*. Blucher.

- Murphy, K. P. (2013). *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.]. ISBN 9780262018029 0262018020.
- Nahmias, S. & Olsen, T. L. (2015). *Production and operations analysis*. Waveland Press.
- Nguyen, D.; Tran, T.; Nguyen, T. & Bose, B. (2008). Wireless broadcast using network coding. *IEEE Transactions on Vehicular technology*, 58(2):914--925.
- Nishida, K. & Yamauchi, K. (2007). Detecting concept drift using statistical testing. Em *International conference on discovery science*, pp. 264--269. Springer.
- oneM2M (2016). oneM2M - Home. <http://www.onem2m.org>. Viewed in Apr 2016.
- Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100--115.
- Pan, Y. & Hu, N. (2014). Research on dependability of cloud computing systems. Em *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)*, pp. 435--439. IEEE.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825--2830.
- Polastre, J.; Szewczyk, R. & Culler, D. (2005). Telos: enabling ultra-low power wireless research. Em *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 364--369. IEEE.
- Qian, F. & Chen, X. (2019). Stock prediction based on lstm under different stability. Em *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp. 483--486. IEEE.
- Qin, F.; Zhang, Q.; Zhang, W.; Yang, Y.; Ding, J. & Dai, X. (2018). Link quality estimation in industrial temporal fading channel with augmented kalman filter. *IEEE Transactions on Industrial Informatics*, 15(4):1936--1946.
- Raza, M.; Ahmed, G. & Khan, N. M. (2012). Experimental evaluation of transmission power control strategies in wireless sensor networks. Em *Emerging Technologies (ICET), 2012 International Conference on*, pp. 1--4. IEEE.

- Renner, C.; Ernst, S.; Weyer, C. & Turau, V. (2011). Prediction accuracy of link-quality estimators. Em *Wireless Sensor Networks*, pp. 1--16. Springer.
- Research, E. (2016). Ettus research product detail - b210. <http://www.ettus.com/product/details/UB210-KIT>. Viewed in Apr 2016.
- Rojas, C. & Decotignie, J.-D. (2018). Leveraging mac preambles for an efficient link estimation. Em *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1--10. IEEE.
- Rumelhart, D. E.; Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533--536.
- Sanches, M. K. (2003). *Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados*. Tese de doutorado, Universidade de São Paulo.
- Senel, M.; Chintalapudi, K.; Lal, D.; Keshavarzian, A. & Coyle, E. J. (2007). A Kalman filter based link quality estimation scheme for wireless sensor networks. Em *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pp. 875--880. IEEE.
- Silva, I.; Guedes, L. A.; Portugal, P. & Vasques, F. (2012). Reliability and availability evaluation of wireless sensor networks for industrial applications. *Sensors*, 12(1):806--838.
- Silva, W. S.; Cordeiro, J. R. S.; Macedo, D. F.; Vieira, M. A.; Vieira, L. F. & Nogueira, J. M. S. (2015). Introdução a rádios definidos por software com aplicações em gnu radio. Em *Minicursos do XXXIII Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos—SBRC*, volume 1, pp. 216--265.
- Sindjoug, M. L. F. & Minet, P. (2019). Wireless link quality prediction in iot networks. Em *PEMWN 2019-8th IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks*.
- Song, L.; Niyato, D.; Han, Z. & Hossain, E. (2015). *Wireless device-to-device communications and networks*. Cambridge University Press.
- Stouffer, K. & Falco, J. (2006). Guide to supervisory control and data acquisition (scada) and industrial control systems security.

- Sun, W.; Lu, W.; Li, Q.; Chen, L.; Mu, D. & Yuan, X. (2017a). Wnn-lqe: Wavelet-neural-network-based link quality estimation for smart grid wsns. *IEEE Access*, 5:12788--12797.
- Sun, W.; Yuan, X.; Wang, J.; Li, Q.; Chen, L. & Mu, D. (2017b). End-to-end data delivery reliability model for estimating and optimizing the link quality of industrial wsns. *IEEE Transactions on Automation Science and Engineering*, 15(3):1127--1137.
- Taleb, T. & Kunz, A. (2012). Machine type communications in 3GPP networks: potential, challenges, and solutions. *Communications Magazine, IEEE*, 50(3):178-184.
- Vallejo, M.; Piorno, J. R. & Rodrigo, J. L. A. (2014). A Link Quality Estimator for Power-Efficient Communication Over On-Body Channels. *2014 12th IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 250--257.
- Verma, P. K.; Verma, R.; Prakash, A.; Agrawal, A.; Naik, K.; Tripathi, R.; Al-sabaan, M.; Khalifa, T.; Abdelkader, T. & Abogharaf, A. (2016). Machine-to-machine (M2M) communications: A survey. *Journal of Network and Computer Applications*, 66:83--105.
- Wan, J.; Chen, M.; Xia, F.; Di, L. & Zhou, K. (2013). From machine-to-machine communications towards cyber-physical systems. *Computer Science and Information Systems*, 10(3):1105--1128.
- Wheelwright, S. C.; Makridakis, S. G. et al. (1973). *Forecasting methods for management*. Wiley.
- Woo, A. & Culler, D. E. (2003). *Evaluation of efficient link reliability estimators for low-power wireless networks*. Computer Science Division, University of California.
- Woo, A.; Tong, T. & Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. Em *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 14--27.
- Xia, N.; Chen, H.-H. & Yang, C.-S. (2018). Radio resource management in machine-to-machine communications—a survey. *IEEE Communications Surveys & Tutorials*, 20(1):791--828.
- Xu, L.; Delaney, D. T.; O'Hare, G. M. & Collier, R. (2013). The impact of transmission power control in wireless sensor networks. Em *Network Computing and*

- Applications (NCA), 2013 12th IEEE International Symposium on*, pp. 255--258. IEEE.
- Xue, X.; Sun, W.; Wang, J.; Li, Q.; Luo, G. & Yu, K. (2020). Rvfl-lqp: Rvfl-based link quality prediction of wireless sensor networks in smart grid. *IEEE Access*, 8:7829--7841.
- Zhang, H.; Sang, L. & Arora, A. (2010). Comparison of data-driven link estimation methods in low-power wireless networks. *Mobile Computing, IEEE Transactions on*, 9(11):1634--1648.
- Zhang, Y.; Yu, R.; Nekovee, M.; Liu, Y.; Xie, S. & Gjessing, S. (2012). Cognitive machine-to-machine communications: Visions and potentials for the smart grid. *IEEE network*, 26(3):6--13.
- Zhou, G.; He, T.; Krishnamurthy, S. & Stankovic, J. A. (2006). Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):221--262.