

Em aplicações de monitoramento, como as Redes de Sensores sem Fio, geralmente são utilizadas grandes quantidades de nós sensores que podem conter fontes de energia limitada e baixa capacidade de memória e processamento. Para enfrentar esses desafios, propõe-se uma abordagem para redução da transmissão de dados em nós sensores baseada na análise da dispersão dos dados coletados. Nos experimentos realizados, foram utilizados conjuntos de dados reais de aplicações de monitoramento publicamente disponíveis. Experimentos realizados no simulador Castalia mostraram que a abordagem proposta obteve resultados promissores quanto à redução das transmissões de dados, mantendo a acurácia dos dados e baixo consumo de energia.

Orientadora: Janine Kniess

Joinville, 2019

ANO
2019

SAMUEL SILVA DE OLIVEIRA | ABORDAGEM PARA REDUÇÃO DE DAOS EM REDES DE SENsoRES SEM FIO BASEADA NA ANÁLISE DA DISPERSÃO DE DADOS

DISSERTAÇÃO DE MESTRADO

**ABORDAGEM PARA REDUÇÃO DE
DADOS EM REDES DE SENsoRES
SEM FIO BASEADA NA ANÁLISE DA
DISPERSÃO DE DADOS**

SAMUEL SILVA DE OLIVEIRA

JOINVILLE, 2019

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA**

SAMUEL SILVA DE OLIVEIRA

**ABORDAGEM PARA REDUÇÃO DE DADOS EM REDES DE
SENSORES SEM FIO BASEADA NA ANÁLISE DE DISPERSÃO DE
DADOS**

JOINVILLE

2019

SAMUEL SILVA DE OLIVEIRA

**ABORDAGEM PARA REDUÇÃO DE DADOS EM REDES DE
SENSORES SEM FIO BASEADA NA ANÁLISE DE DISPERSÃO DE
DADOS**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dra. Janine Kniess

JOINVILLE

2019

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Setorial do CCT/UDESC,
com os dados fornecidos pelo(a) autor(a)**

Oliveira, Samuel Silva de
Abordagem Para Redução de Dados em Redes de
Sensores sem Fio Baseada na Análise da Dispersão de
Dados / Samuel Silva de Oliveira. -- 2019.
123 p.

Orientadora: Janine Kniess
Dissertação (mestrado) -- Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas, Programa
de Pós-Graduação , Joinville, 2019.

1. Redes de Sensores sem Fio. 2. Redução de Dados. 3.
Análise da Dispersão. 4. Internet das Coisas. I. Kniess,
Janine. II. Universidade do Estado de Santa Catarina, Centro
de Ciências Tecnológicas, Programa de Pós-Graduação . III.
Título.

**Abordagem Para Redução de Dados em Redes de Sensores Sem Fio Baseado
na Análise de Dispersão de Dados**

por

Samuel Silva de Oliveira

Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada

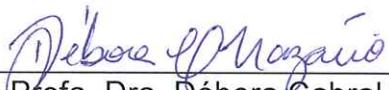
Área de concentração em “Ciência da Computação”,
e aprovada em sua forma final pelo

CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

Banca Examinadora:

Profa. Dra. Janine Kniess
CCT/UDESC (Orientadora/Presidente)

Prof. Dr. José Viterbo Filho
UFF


Profa. Dra. Débora Cabral Nazário
CCT/UDESC

Joinville, SC, 25 de março de 2019.

AGRADECIMENTOS

À Deus, por conduzir os meus passos, me concedendo ânimo para lidar com todas as dificuldades.

À minha família, em especial à minha esposa Tháfinys e minha filha Alice por terem me acompanhado durante esses dois anos, sempre demonstrando total apoio, carinho e compreensão.

À meus pais, Paulo e Marilene, que durante toda a minha vida me apoiaram e me deram o suporte necessário para que eu pudesse chegar até aqui.

À minha orientadora, Prof^a. Dra. Janine Kniess, que desde o início desta pesquisa me orientou com toda dedicação, oferecendo direcionamentos e também, fazendo as cobranças necessárias para que o trabalho fosse concluído com êxito.

À todos professores do Programa de Pós-Graduação em Computação Aplicada por todo o conhecimento compartilhado, contribuindo para o meu crescimento profissional.

Obrigado!

“Não é tolo aquele que abre mão do que não pode reter, para ganhar o que não pode perder.”

Jim Elliot

RESUMO

Em aplicações de monitoramento, como as Redes de Sensores sem Fio (RSSF), dispositivos chamados de nós sensores são responsáveis por captar informações do ambiente físico e enviá-las para o seu destinatário. Nessas aplicações, geralmente são utilizados grandes quantidades de nós sensores que podem conter fontes de energia limitada e baixa capacidade de memória e processamento. Além disso, em aplicações com grande número de sensores, a sobrecarga na comunicação pode ser muito alta. Para enfrentar esses desafios, propõe-se uma abordagem para redução da transmissão de dados em nós sensores baseada na análise da dispersão dos dados coletados. Essa abordagem visa evitar o envio de detecções cujo os valores apresentem baixa dispersão. Ou seja, as transmissões só são realizadas quando há variação significante no ambiente monitorado. A abordagem proposta conta com um método de reconstrução dos dados, cujo objetivo é reconstruir a série temporal do sensor baseado na tendência assintótica dos valores que deixaram de ser enviados, dada a sua baixa dispersão. Através do uso desta abordagem, obtém-se um conjunto de dados final a partir de um conjunto de dados inicial com menor quantidade de detecções. Além disso, a acurácia dos dados reconstruídos é avaliada através da análise dos modelos de predição *Recurrent Neural Network* (RNN), *Auto-Regressive Integrated Moving Average* (ARIMA) e *Support Vector Machine* (SVM). Nos experimentos realizados, foram utilizados conjuntos de dados reais de aplicações de monitoramento publicamente disponíveis. A abordagem proposta foi comparada com os métodos *Adaptive Method for Data Reduction* (AMDR), *Water Meter Readings Prediction* (WMRP) e *Quality-Based Data Reduction* (QBDR). Experimentos realizados no simulador Castalia, mostraram que a abordagem proposta obteve resultados promissores quanto à redução das transmissões de dados, mantendo a acurácia dos dados e baixo consumo de energia.

Palavras-chaves: Redes de Sensores sem Fio. Redução de Dados. Análise da Dispersão. Internet das Coisas

ABSTRACT

In monitoring applications, such as Wireless Sensor Networks (WSN), sensor nodes are responsible for sensing information from the physical environment and transmitting it to its destiny. In these applications, large amounts of sensor nodes are usually used and they may contain limited power sources, as well as low memory and processing capabilities. In addition, in applications with large numbers of sensors, communication overhead can be very high. To address these challenges, we propose an approach to reduce data transmissions in sensor nodes based on the sensed data dispersion analysis. This approach aims to avoid transmitting measurements whose values demonstrate low dispersion. That is, the transmissions are only performed when there is significant variation in the sensed environment. The proposed approach contains a data reconstruction method, which aims to reconstruct the sensor time series based on the asymptotic trend from the data which were not transmitted, given its low dispersion. Thus, it is expected to reconstruct a final data set from an initial reduced data set. In addition, the accuracy of the reconstructed data is evaluated through the analysis of the prediction models *Recurrent Neural Network* (RNN), *Auto-Regressive Integrated Moving Average* (ARIMA) and *Support Vector Machine* (SVM). In the performed experiments, real data sets of publicly available monitoring applications were used. The proposed approach was compared with the methods *Adaptive Method for Data Reduction* (AMDR), *Water Meter Readings Prediction* (WMRP) and *Quality-Based Data Reduction* (QBDR). The experiments carried out with Castalia Simulator shows that the results obtained by the proposed approach were promising for reducing data transmissions, while maintaining data accuracy and low energy consumption.

Key-words: Wireless Sensor Networks. Data Reduction. Dispersion Analysis. Internet of Things.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Topologias de RSSF: (a) Ponto-a-ponto; (b) Estrela; (c) Árvore. Adaptado de (REINA et al., 2013)	32
Figura 2.2 – Classificação de Técnicas para Redução de Dados em RSSF. Fonte: Próprio Autor	33
Figura 2.3 – Abordagens de Agregação em RSSF: (a) Árvore; (b) Cluster	34
Figura 2.4 – Fluxo de Execução do Esquema de Predição Dupla. Fonte: Próprio Autor	38
Figura 2.5 – Esquema de Predição Dupla: Compartilhamento do Algoritmo de Predição. Adaptado de (DIAS; BELLALTA; OECHSNER, 2016a)	39
Figura 2.6 – Utilização da Técnica de CS em RSSF. Adaptado de (XIE; JIA, 2014)	40
Figura 2.7 – Exemplo de Dispersão nos Dados. Fonte: Próprio Autor	41
Figura 2.8 – Separação por Máxima Margem Linear. Fonte: Adaptado de (ALBUQUERQUE, 2014)	45
Figura 2.9 – Funções de <i>kernel</i> lineares e não-lineares do modelo SVR. Fonte: (SCIKIT-LEARN, 2018)	47
Figura 2.10 – Rede Neural Artificial. Fonte: Adaptado de (DERTAT, 2017)	48
Figura 2.11 – Neurônio da Camada Oculta em uma RNN. Fonte: (OLAH, 2014)	49
Figura 4.1 – Fluxo de execução do Algoritmo de Reconstrução	65
Figura 4.2 – (a) Série original; (b) Série manipulada, onde $D_{max} = 0,4$; (c) Série manipulada, onde $D_{max} = 0,7$. Fonte: Próprio Autor	66
Figura 4.3 – Tendência Assintótica. Fonte: Próprio Autor	68
Figura 5.1 – Percentual de Redução na Transmissão de Dados. Fonte: Próprio Autor	84
Figura 5.2 – Erro Absoluto Médio (EAM). Fonte: Próprio Autor	85
Figura 5.3 – Erro Absoluto Médio (EAM) obtido nos modelos RNN, ARIMA e SVM. Fonte: Próprio Autor	90
Figura 5.4 – R^2 : Modelos RNN, ARIMA e SVM. Fonte: Próprio Autor	91
Figura 5.5 – Tempo de Execução: Modelos RNN, ARIMA e SVM. Fonte: Próprio Autor	92
Figura 5.6 – Percentual Médio de Transmissões de Dados. Fonte: Próprio Autor	95
Figura 5.7 – Percentual Médio de Consumo de Energia. Fonte: Próprio Autor	95
Figura 5.8 – Erro Absoluto Médio (EAM). Fonte: Próprio Autor	97
Figura 5.9 – Atraso na Entrega das Mensagens. Fonte: Próprio Autor	98
Figura 5.10 – Atraso na Entrega das Mensagens Otimizadas. Fonte: Próprio Autor	98
Figura 5.11 – Tempo de Execução. Fonte: Próprio Autor	100
Figura 5.12 – Tempo de Execução. Fonte: Próprio Autor	100

Figura 8.1 – Quantidade de Transmissões. Fonte: Próprio Autor	109
Figura 8.2 – Consumo de Energia dos Nós Sensores. Fonte: Próprio Autor . . .	110

LISTA DE TABELAS

Tabela 3.1 – Comparação dos Trabalhos Relacionados	61
Tabela 4.1 – Parâmetros e Variáveis utilizados nos Algoritmos 4.1, 4.2 e 4.3	70
Tabela 5.1 – Detalhes dos <i>Datasets</i> Utilizados	79
Tabela 5.2 – Análise da Redução e Reconstrução de Dados (EAM)	82
Tabela 5.3 – Análise da Predição de Valores Futuros	89
Tabela 5.4 – Parâmetros de Simulação	94
Tabela 7.1 – Resultados obtidos no Castalia	107

LISTA DE SIGLAS E ABREVIATURAS

AMDR *Adaptive Method for Data Reduction*

ARIMA *Auto-Regressive Integrated Moving Average*

AS-TCP *Adaptive Sampling - Transmission Control Protocol*

BRKGA *Biased Random Key Genetic Algorithm*

CER *Continuous Enhancement Routing Solution*

CS *Compressive Sensing*

CWA *Confidence-Weighted Averaging*

DAARP *Data Aggregation Aware Routing Protocol*

DCT *Discrete Cosine Transform*

DDAARP *Dynamic Data Aggregation Aware Routing Protocol*

DES *Double Exponential Smoothing*

DP Desvio Padrão

DPCAS *Dual Prediction with Cubic Adaptive Sampling*

DPS *Dual Prediction Scheme*

DTCS *Dynamic Topology Compression Sensing*

EAM Erro Absoluto Médio

EDSAS *Double Smoothing-based Adaptive Sampling*

GPS *Global Positioning System*

IoT *Internet of Things*

LMS *Least Mean Square*

M2M *Machine-to-machine*

MEMS *Micro Electro Mechanical Systems*

NDBC *Nation Data Buoy Center*

NFC *Near Field Communication*

QBDR *Quality-Based Data Reduction*

RFID *Radio Frequency Identification*

RIP *Restricted Isometry Property*

RNN *Recurrent Neural Network*

RSSF *Redes de Sensores sem Fio*

SPT *Shortest Path Tree*

SVM *Support Vector Machine*

SVR *Support Vector Regression*

WMRP *Water Meter Readings Prediction*

LISTA DE SÍMBOLOS

$\%$	Porcentagem
$D_{v_i, v_{i-1}}$	Distância de Manhattan
\emptyset	Conjunto vazio
R^2	Coeficiente de Determinação
\leq	Menor ou igual
\geq	Maior ou igual
$=$	Igual
\mathbb{R}	Conjunto dos números reais
Σ	Somatório
\subset	Está contido
\in	Pertence
\aleph	Números <i>Aleph</i>
\pm	Aproximadamente

SUMÁRIO

1	INTRODUÇÃO	23
1.1	OBJETIVOS	25
1.2	METODOLOGIA	26
1.3	ORGANIZAÇÃO DO TRABALHO	27
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	INTERNET DAS COISAS	29
2.2	REDES DE SENsoRES SEM FIO	31
2.3	MÉTODOS PARA REDUÇÃO DE DADOS EM RSSF	33
2.3.1	Técnicas Globais de Redução de Dados em RSSF (Topológicas)	33
2.3.1.1	<i>Agregação de Dados</i>	34
2.3.1.2	<i>Fusão de Dados</i>	36
2.3.2	Técnicas Locais de Redução de Dados em RSSF (Não-Topológicas)	37
2.3.2.1	<i>Predição de Dados</i>	37
2.3.2.2	<i>Compressive Sensing (CS)</i>	39
2.3.2.3	<i>Análise da Dispersão</i>	41
2.4	PREDIÇÃO DE SÉRIES TEMPORAIS	43
2.4.1	<i>Auto-Regressive Integrated Moving Average (ARIMA)</i>	43
2.4.2	<i>Support Vector Machine (SVM)</i>	44
2.4.3	Redes Neurais Artificiais	47
2.5	CONCLUSÕES DO CAPÍTULO	49
3	TRABALHOS RELACIONADOS	51
3.1	MÉTODOS BASEADOS EM AGREGAÇÃO DE DADOS	51
3.2	MÉTODOS BASEADOS EM FUSÃO DE DADOS	53
3.3	MÉTODOS BASEADOS EM PREDIÇÃO DE DADOS	54
3.4	MODELOS BASEADOS EM <i>COMPRESSIVE SENSING</i>	58
3.5	COMPARAÇÃO DOS TRABALHOS RELACIONADOS	60
3.6	CONCLUSÕES DO CAPÍTULO	62
4	ABORDAGEM PARA REDUÇÃO DAS TRANSMISSÕES DE DADOS EM RSSF	63
4.1	ANÁLISE DA DISPERSÃO DE VALORES DETECTADOS EM NÓS SENSORES	63
4.2	ALGORITMO PARA RECONSTRUÇÃO DOS DADOS	66

4.3	FASE DE INICIALIZAÇÃO PARA A ESCOLHA DO VALOR DE D_{max}	68
4.4	MODELOS DE PREDIÇÃO DE SÉRIES TEMPORAIS	73
4.4.1	Modelo <i>Auto-Regressive Integrated Moving Average (ARIMA)</i>	73
4.4.2	Modelo <i>Support Vector Regression (SVR)</i>	73
4.4.3	Modelo baseado em Rede Neural Recorrente	74
4.4.4	Métricas para Avaliação dos Valores Previstos	75
4.5	CONCLUSÕES DO CAPÍTULO	76
5	EXPERIMENTOS E RESULTADOS	77
5.1	CONJUNTOS DE DADOS AVALIADOS	77
5.2	ANÁLISE DA REDUÇÃO E RECONSTRUÇÃO DOS DADOS	80
5.2.1	Testes de Significância Estatística	85
5.3	AVALIAÇÃO DA PREDIÇÃO DE VALORES FUTUROS	86
5.4	RESULTADOS COMPUTACIONAIS NO CASTALIA	92
5.5	CONCLUSÕES DO CAPÍTULO	101
6	CONCLUSÃO E TRABALHOS FUTUROS	103
6.1	TRABALHOS FUTUROS	105
6.2	OUTRAS CONTRIBUIÇÕES DA DISSERTAÇÃO	105
7	APÊNDICE A - RESULTADOS OBTIDOS NO CASTALIA	107
8	APÊNDICE B - CONSIDERAÇÕES SOBRE A REDUÇÃO DE DADOS E CONSUMO DE ENERGIA NO CASTALIA	109
	REFERÊNCIAS	111

1 INTRODUÇÃO

O avanço da tecnologia possibilitou que a Internet pudesse deixar de ser um conjunto de redes de computadores, para expandir-se a um novo contexto: a Internet das Coisas¹. Nesse novo cenário, objetos da vida cotidiana das pessoas passam a ter capacidade computacional e de comunicação, podendo conectar-se à Internet, unindo o mundo físico ao mundo da informação (SANTOS et al., 2016).

Gubbi et al. (2013) oferecem um novo conceito para Internet das Coisas: uma interconexão de dispositivos de detecção e atuação que oferecem capacidade de compartilhar informações em plataformas através de uma estrutura unificada, desenvolvendo uma imagem operacional comum para permitir aplicações inovadoras. Esse conceito deixa claro que através da Internet das Coisas é possível captar informações sem a necessidade de interação humana e também realizar inúmeras tarefas com autonomia.

Nas aplicações de Internet das Coisas (IoT) focadas em monitoramento, os nós sensores desempenham uma função essencial: monitoramento e captação de informações. Consequentemente, a utilização desses dispositivos permite melhorias significativas nos métodos de sensoriamento, podendo ser implantados de maneira interligada, formando uma Redes de Sensores sem Fio (RSSF).

Nas Redes de Sensores sem Fio, como aquelas utilizadas em IoT, os nós sensores realizam detecções que geralmente são ordenadas pelo tempo, no formato de séries temporais. Segundo Morettin e Toloi (1981), uma série temporal pode ser definida como sendo qualquer conjunto de observações ordenadas no tempo. Os nós sensores geralmente possuem poucos recursos, como memória, processamento e capacidade energética. Por isso, é comum que eles sejam programados para enviar os dados coletados de forma imediata para as estações bases, chamadas de nós sorvedouro², que são responsáveis pelo encaminhamento das mensagens até o seu destino final. Uma vez que a quantidade dos nós sensores em atividade é expressiva, a transmissão de dados é um fator dominante para a sobrecarga na comunicação, bem como para o consumo de energia.

Alsheikh et al. (2015) destaca que nas Redes de Sensores Sem Fio, até 80% do consumo total de energia pode estar relacionado às atividades de transmissões de dados. Com base no exposto, é possível identificar que a minimização das transmissões de dados dos sensores nas RSSF podem prolongar o seu tempo de vida.

¹ Do Inglês: *Internet of Things* (IoT)

² Em Inglês: Sink nodes

Contudo, Alsheikh et al. (2016) destaca que a redução das transmissões de dados a partir dos sensores não apenas resulta em economia de energia, mas também na redução do consumo na largura de banda da rede, possibilitando o uso mais eficiente dos recursos disponíveis.

Segundo Gartner (2017), 20 bilhões de objetos inteligentes devem estar conectados à Internet das Coisas até 2020, superando a população mundial em termos demográficos. Isso significa que o volume de dados gerado pelos dispositivos de IoT será cada vez maior, demandando o desenvolvimento de novas estratégias que prolonguem a sua disponibilidade.

Neste trabalho, é proposto um método para redução da transmissão de dados a partir de nós sensores atuando em aplicações de monitoramento, como nas Redes de Sensores sem Fio. O método é baseado na análise da dispersão dos dados detectados, cujo objetivo é identificar conjuntos de detecções com valores que demonstrem baixa dispersão. Justifica-se a utilização deste método, pois é comum que as detecções realizas por nós sensores, por vezes, sejam muito semelhantes, ou até mesmo idênticas, como afirmam os autores El-Telbany e Maged (2017).

Para exemplificar, dado o conjunto de detecções de um sensor de temperatura $T = \{12,2, 12,4, 12,3, 12,2, 12,7, 14,7, 16,0, 18,9\}$, pode-se constatar que as quatro primeiras detecções de T são menos dispersas, pois seus valores são mais próximos uns dos outros. Com o uso do método proposto, ao invés de enviar todas as detecções de T , as quatro primeiras detecções seriam suprimidas, e apenas uma transmissão seria realizada referente aos valores pouco dispersos. Nesse exemplo, embora T possua oito detecções, apenas cinco transmissões seriam realizadas. Através dessa redução, pretende-se utilizar de forma mais eficiente os recursos da rede, como largura de banda e energia dos nós. O método proposto conta ainda com um algoritmo de reconstrução de dados, que visa fornecer uma aproximação dos valores detectados que deixaram de ser enviados pelos sensores, garantindo um certo nível de acurácia com relação aos dados originais.

Neste trabalho de dissertação, o foco está nas aplicações de monitoramento nas quais os dados coletados são numéricos. Dias (2016) destaca que aplicações que utilizam sensores para monitoramento geralmente coletam dados numéricos (nominais, ordinais, intervalos ou proporções) e podem ser representados de diversas formas. Por exemplo, distâncias podem ser representadas em centímetros ou metros. Temperaturas podem ser representadas em *Kelvin, Fahrenheit ou Celsius*. O tempo, contudo, pode ser representado em minutos, horas ou dias. Dados de temperatura, umidade, distância e vibração são exemplos de informações detectadas através de sensores distribuídos em diversos tipos de aplicações no contexto da Internet das Coisas. Ressalta-se que a abordagem proposta é genérica e pode ser utilizada em

aplicações de monitoramento atuando fora do contexto de IoT. Contudo, no âmbito deste trabalho, a ênfase está nos dados capturados através de aplicações no contexto da Internet das Coisas.

As aplicações que utilizam sensores para fins de monitoramento, geralmente dão origem a coleções de dados ordenados pelo tempo, no formato de séries temporais. Dessa forma, é comum que informações históricas sejam utilizadas para realizar a previsão de valores futuros. De acordo com Shi-Nash e Hardoon (2016), a análise preditiva pode ser fundamental em aplicações de monitoramento, a fim de prever condições futuras do ambiente sensoreado. Nos trabalhos (VERHEGGHEN et al., 2012; SILVA, 2008; SCOTTA, 2012) é possível constatar a importância da análise de séries temporais para diversas aplicações de monitoramento.

Com base no exposto, este trabalho também contempla a análise da acurácia de dados gerados pelo algoritmo de reconstrução através de modelos de previsão de séries temporais, amplamente utilizados no contexto de aplicações de monitoramento através de sensores. Dessa forma, foram analisados três modelos de previsão de séries temporais encontrados na literatura. São eles:

- Modelo Auto-Regressivo Integrado de Médias Móveis;
Em Inglês: *Auto-Regressive Integrated Moving Average* (ARIMA);
- Máquina de Vetores de Suporte;
Em Inglês: *Support Vector Machine* (SVM);
- Redes Neurais Recorrentes.
Em Inglês: *Recurrent Neural Network* (RNN).

A análise e comparação destes modelos foi realizada com o intuito de identificar qual dos modelos alcança maior acurácia na previsão dos dados coletados por sensores, bem como, aferir se a reconstrução dos dados, decorrente da abordagem proposta para redução das transmissões de dados, afeta a previsão de valores futuros.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma abordagem para a redução da quantidade de transmissões de dados realizadas por nós sensores em aplicações de monitoramento no contexto da Internet das Coisas. Através do uso desta abordagem é possível alcançar um consumo mais eficiente dos recursos da rede, como largura de banda e energia dos nós sensores.

As principais contribuições deste trabalho incluem:

- Um algoritmo de análise da dispersão de detecções, responsável por identificar períodos estacionários (de baixa dispersão) nos valores detectados pelos sensores e então encaminhar ao sorvedouro apenas uma mensagem correspondente ao período inteiro, contendo valores específicos a serem utilizados na fase de reconstrução;
- Um algoritmo para a reconstrução dos dados, cujo objetivo é reconstruir a série temporal do sensor baseado na tendência assintótica dos valores que deixaram de ser enviados;
- Investigação, análise e comparação dos modelos de previsão de séries temporais *Auto-Regressive Integrated Moving Average* (ARIMA), *Support Vector Machine* (SVM) e *Recurrent Neural Network* (RNN), a fim de identificar qual dos modelos alcança maior acurácia na previsão de valores futuros, bem como, constatar se há interferência da fase de reconstrução de dados com relação à previsão de valores futuros.

1.2 METODOLOGIA

Para o desenvolvimento da abordagem proposta neste trabalho de dissertação, foi realizada uma pesquisa exploratória baseada em dois principais objetivos: (i) conhecer o estado-da-arte com relação à aplicação de técnicas para a redução das transmissões de dados em aplicações de monitoramento baseadas em sensores, como Redes de Sensores sem Fio; (ii) investigar a utilização de modelos de previsão de séries temporais encontrados na literatura.

Posteriormente, a abordagem proposta foi projetada, incluindo a implementação do algoritmo de redução de dados, baseado na análise da dispersão de detecções e do algoritmo de reconstrução de dados baseado na tendência assintótica dos conjuntos de detecções com baixa dispersão. Por fim, foi realizada a implementação de modelos de previsão de séries temporais.

Após a construção da abordagem proposta e implementação dos modelos de previsão de séries temporais, foram realizados diversos experimentos com o intuito de verificar e validar a eficiência e desempenho dos mesmos. Para isto, foram utilizados conjuntos de dados reais provenientes de aplicações de sensores de monitoramento, publicamente disponíveis.

Os experimentos realizados tiveram como objetivo a avaliação do potencial da abordagem proposta com relação às seguintes métricas: quantidade de transmissões realizadas, acurácia dos dados reconstruídos, e previsão de valores futuros (dados reais e reconstruídos). Além disso, foram realizadas simulações com o simulador Cas-

talia (RASTEGARNIA; SOLOUK, 2011) a fim de avaliar a quantidade de transmissões, consumo de energia dos nós sensores, atraso na entrega das mensagens, tempo de processamento do algoritmo de reconstrução e a acurácia dos dados reconstruídos.

Para fins de comparação, foram implementados trabalhos encontrados na literatura que lidam com este problema. São eles: *Adaptive Method for Data Reduction* (AMDR) (FATHY; BARNAGHI; TAFAZOLLI, 2018), *Water Meter Readings Prediction* (WMRP) (ALMHANA; CHOULAKIAN; ALMHANA, 2017) e *Quality-Based Data Reduction* (QBDR) (SANTINI; ROMER, 2006).

1.3 ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho está organizado da seguinte forma: O Capítulo 2 descreve a fundamentação teórica, detalhando os principais conceitos, definições e métodos relacionados à abordagem proposta, além de envolver os seguintes temas: Internet das Coisas, Redes de Sensores sem Fio e Predição de Séries Temporais. No Capítulo 3, trabalhos relacionados à presente pesquisa são apresentados. No Capítulo 4, descrevem-se as implementações dos algoritmos de análise da dispersão dos dados detectados pelos sensores e dos modelos de predição de séries temporais. No Capítulo 5, os resultados encontrados são apresentados e discutidos, e por fim, no Capítulo 6 apresentam-se as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresenta-se uma revisão de literatura como fundamentação teórica deste trabalho. Primeiramente, a Seção 2.1 aborda sobre o tema Internet das Coisas. Na Seção 2.2 apresentam-se conceitos e definições sobre Redes de Sensores Sem Fio e sua relação com o problema da redução de dados. Na Seção 2.3, são apresentados os métodos do estado-da-arte para redução de dados em redes que utilizam sensores e atuam em aplicações de monitoramento. Na Seção 2.4 são apresentados conceitos sobre previsão de séries temporais, além de algoritmos e modelos de predição encontrados na literatura que lidam com este objetivo. Por fim, na Seção 2.5 são apresentadas as conclusões do capítulo.

2.1 INTERNET DAS COISAS

A Internet das Coisas é um tema que vem despertando a atenção de diversos setores nos últimos anos. A ideia principal consiste na presença de objetos interligados e com acesso à Internet. Tais objetos possuem capacidades de processamento e comunicação, como por exemplo, etiquetas de identificação por rádio-frequência¹, sensores e atuadores, além de poder interagir uns com os outros a fim de alcançar objetivos comuns (GUBBI et al., 2013).

O termo Internet das Coisas foi primeiramente mencionado por Kevin Ashton em 1999, enquanto fazia uma apresentação sobre o uso da tecnologia *Radio Frequency Identification* (RFID) em um sistema de gestão de suprimentos conectado à Internet (ASHTON, 2009). Desde então esse tema tem ganhado destaque, não limitando-se apenas ao RFID, mas envolvendo diversos padrões de comunicação, como WiFi (STALLINGS, 2004), *Near Field Communication* (NFC) (WANT, 2011), *ZigBee* (LIU, 2014), *Bluetooth Low Energy* (HUGHES; YAN; SOGA, 2015), *Long Range Wide Area Network* (LoRaWAN) (FEHRI et al., 2018), *SigFox* (MEKKI et al., 2018), dentre outros.

Cada padrão de comunicação possui características específicas, como por exemplo: faixa de frequência, latência, consumo de energia, raio de alcance e outros. Segundo Pule, Yahya e Chuma (2017), não há uma solução de conectividade única considerada adequada para todas as aplicações de IoT. A escolha do padrão depende inteiramente dos requisitos de comunicação e das restrições de recursos de uma determinada aplicação.

¹ Do Inglês: *Radio Frequency Identification* (RFID)

A Internet das Coisas tem se mostrado promissora em diversas áreas, como meio ambiente, saúde, transporte, energia, e outros. A evolução tecnológica nas áreas de sistemas embarcados, microeletrônica, comunicação e sensoriamento proporcionou a criação de novos objetos inteligentes que fazem com que a Internet das Coisas seja cada vez mais possível (SANTOS et al., 2016).

De acordo com Lee e Lee (2015), cinco tecnologias são cruciais para o desenvolvimento de soluções baseadas em Internet das Coisas. São elas: Tecnologia *Radio-Frequency Identification* (RFID) que utiliza etiquetas inteligentes a fim de permitir a identificação automática de objetos; Redes de Sensores sem Fio (RSSF), que são conjuntos de pequenos dispositivos inteligentes com capacidades de sensoriamento, processamento e comunicação; Middleware, que é um componente de *software* que atua como uma interface entre aplicações a fim de facilitar a comunicação entre elas; Computação em nuvem, que se refere a recursos de hardware e aplicações em Data-centers provendo serviços na Internet, e por fim, Comunicação *Machine-to-machine* (M2M)² utilizada para interação entre dispositivos inteligentes.

Manrique, Rueda-Rueda e Portocarrero (2016) destacam que as RSSF podem ser consideradas como sendo uma parte integrante da Internet das Coisas, por realizar a interconexão de objetos inteligentes, chamados de nós sensores. Os nós sensores em uma RSSF são pequenos dispositivos que possuem três componentes básicos: detecção, processamento e comunicação. O componente de detecção é responsável pela coleta de informações do ambiente físico. O componente de processamento realiza processamento e armazenamento de dados localmente. O componente de comunicação, por sua vez, é responsável pela transmissão dos dados coletados a uma estação-base. Além disso, cada nó sensor deve dispor de uma fonte de energia para o seu funcionamento. Esta fonte de energia geralmente consiste em uma bateria com uma quantidade limitada de energia.

Uma vez que as aplicações de IoT que utilizam sensores são geralmente densas, ou seja, possuem grande número de objetos, os dados gerados por essas aplicações são massivos, podendo aumentar a sobrecarga da rede, bem como consumir demasiadamente a energia dos dispositivos. Dessa maneira, técnicas voltadas para a redução das transmissões de dados se fazem necessárias para que os recursos da rede sejam utilizados com mais eficiência e para que o tempo de vida da rede seja estendido.

Através do esforço colaborativo dos nós sensores em uma RSSF, é possível alcançar os objetivos gerais da rede, como por exemplo: monitorar variáveis do clima de uma cidade. Na próxima seção será abordado sobre o tema de Redes de Sensores sem Fio e suas características.

² Em Português: Máquina-a-máquina

2.2 REDES DE SENSORES SEM FIO

De acordo com Yick, Mukherjee e Ghosal (2008), a proliferação das tecnologias de sistemas microeletromecânicos³, comunicação sem fio e eletrônica digital impulsionaram o desenvolvimento de dispositivos de baixo custo com capacidades de sensoriamento, processamento e comunicação sem fio, normalmente chamados de nós sensores. A utilização destes dispositivos em conjunto caracteriza a formação de Redes de Sensores sem Fio, que são baseadas no esforço colaborativo de uma grande quantidade de nós.

Existem aplicações de RSSF na área da agricultura de precisão (LI; CUI; LI, 2011), na manutenção preditiva de equipamentos (GOMBÉ et al., 2017), monitoramento da qualidade da água (ADU-MANU et al., 2017) e muitos outros. Yick, Mukherjee e Ghosal (2008) destacam que as aplicações mais comuns de monitoramento em RSSF são: monitoramento ambiental interno e externo (temperatura, umidade), monitoramento de saúde e bem-estar, monitoramento de energia, monitoramento de localização de inventário e monitoramento sísmico e estrutural.

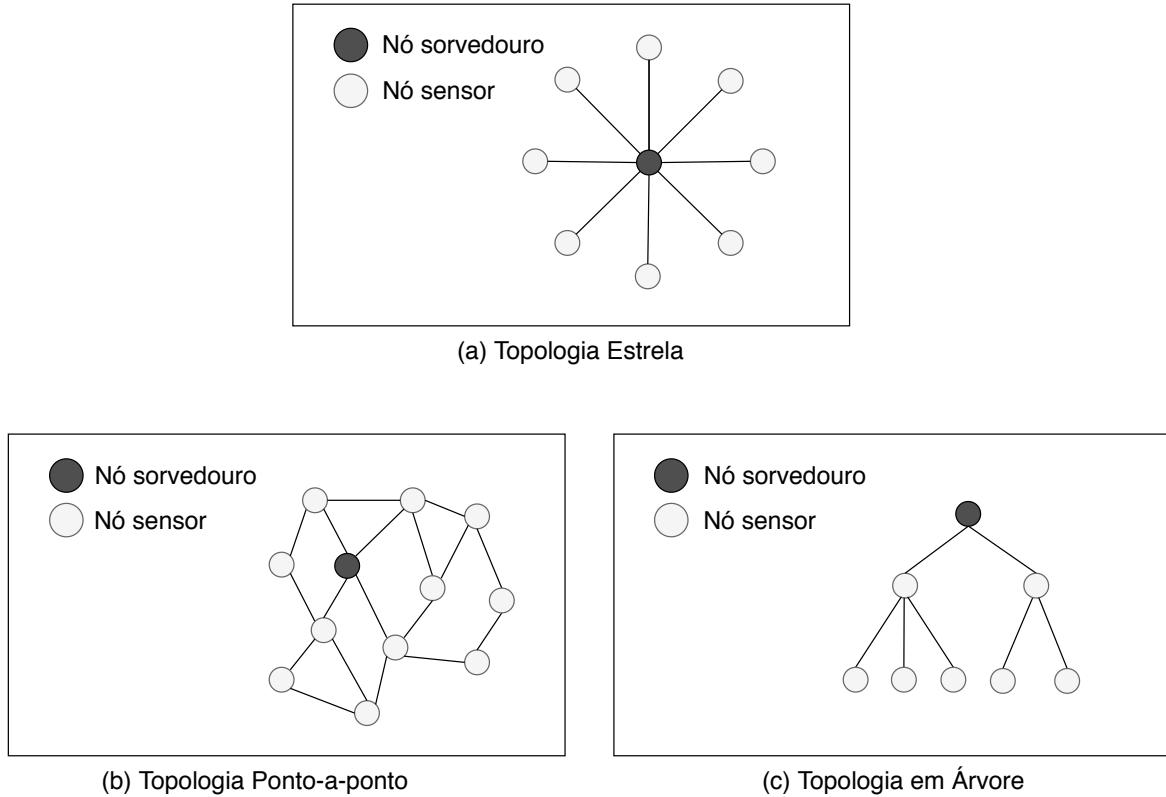
As Redes de Sensores sem Fio são compostas por um conjunto de nós sensores interligados (normalmente centenas ou milhares) e geralmente apresentam fontes de energia limitadas. A principal tarefa dos nós sensores em RSSF é realizar a coleta de dados sobre variáveis do ambiente onde estão implantados e enviá-los à uma estação-base, chamada de nó sorvedouro. As RSSF não dependem de uma infraestrutura pré-existente, como roteadores em redes cabeadas ou pontos de acesso (ZANJIREEH; SHAHRABI; LARIJANI, 2013). Ao invés disso, elas possuem um ou mais nós sorvedouro, que são responsáveis por receber os dados provenientes das detecções dos sensores e encaminhá-los para o seu destino final, como por exemplo, um servidor na nuvem.

A maneira como os sensores interagem entre si depende da topologia adotada para a rede. Segundo Reina et al. (2013), Redes de Sensores sem Fio podem ser implantadas em três principais topologias: estrela, árvore e ponto-a-ponto, como ilustrado na Figura 2.1.

Na topologia em estrela, os nós são posicionados a um único salto de distância do nó sorvedouro. Segundo Matin e Islam (2012), esse modelo de topologia é caracterizado pela simplicidade, pelo baixo consumo de energia dos nós sensores e pelo baixo atraso na comunicação entre nós sensores e nó sorvedouro. Porém, a desvantagem dessa topologia é que o nó sorvedouro precisa estar no raio de alcance de todos os nós sensores da rede, além de não ser tão robusta quanto as outras topologias devido à sua dependência em um único nó para gerenciar a rede.

³ Do Inglês: *Micro Electro Mechanical Systems* (MEMS)

Figura 2.1 – Topologias de RSSF: (a) Ponto-a-ponto; (b) Estrela; (c) Árvore. Adaptado de (REINA et al., 2013)



Nas topologias ponto-a-ponto e árvore, as comunicações são realizadas através de múltiplos saltos. Ou seja, se um determinado nó sensor estiver fora do raio de alcance do nó sorvedouro, ele poderá usar os nós sensores intermediários para encaminhar seus dados. Nesses casos, é necessária a definição de rotas para que os dados dos sensores cheguem até ao nó sorvedouro, uma vez que diversos caminhos podem ser adotados. A criação de rotas baseada na quantidade de energia dos nós sensores intermediários é uma abordagem bastante utilizada. Vários protocolos e algoritmos de roteamento para Redes de Sensores sem Fio foram propostos em (LIU; QIAN, 2016), (RAO; REDDY, 2017), (ELSHRKAWEY; ELSHERIF; WAHED, 2018), (GUPTA; JHA, 2018) e (ARASU; GANESAN, 2018).

Como as RSSF costumam possuir grande número de nós, a quantidade de dados trafegada na rede pode ser significativa em relação aos recursos disponíveis nos sensores, e requerem que técnicas sejam utilizadas a fim de diminuir a quantidade de transmissões na rede, mantendo a confiabilidade dos dados detectados.

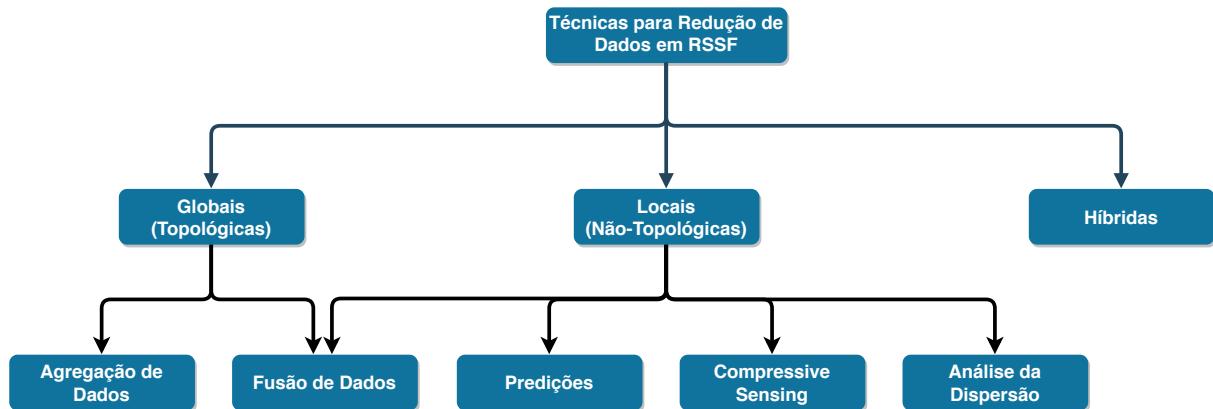
Na próxima seção serão abordadas algumas técnicas encontradas na literatura, que lidam com a redução de dados em Redes de Sensores sem Fio.

2.3 MÉTODOS PARA REDUÇÃO DE DADOS EM RSSF

De acordo com Elshrkawey, Elsherif e Wahed (2018), o consumo de energia desempenha um papel importante em Redes de Sensores sem Fio, cujo os nós da rede não tem uma fonte de energia disponível, uma vez que a energia consumida na transmissão de dados a partir de nós sensores pode ser alta, dependendo do número de transmissões realizadas. É importante ressaltar que, dependendo da topologia da rede, como a topologia em árvore e topologia ponto-a-ponto, os sensores não transmitem apenas as suas detecções, mas também encaminham os dados de outros sensores, aumentando ainda mais o número de transmissões que um sensor pode realizar. Como consequência desta característica da rede, a sobrecarga dos nós aumenta, assim como, o consumo de energia dos dispositivos.

Anastasi et al. (2009) propõe uma classificação para técnicas de redução de dados em Redes de Sensores sem Fio (RSSF), ilustrada na Figura 2.2. As técnicas nas quais a redução acontece após o envio dos dados do nó sensor para a rede são chamadas de técnicas globais, ou topológicas. As técnicas que realizam a redução no nó sensor, antes dos dados serem enviados para a rede são chamadas de técnicas locais (não-topológicas). Existem ainda técnicas híbridas que utilizam os conceitos de ambas classificações (globais e locais).

Figura 2.2 – Classificação de Técnicas para Redução de Dados em RSSF. Fonte: Próprio Autor



A seguir serão apresentadas técnicas do estado-da-arte para para redução de dados em Redes de Sensores sem Fio (RSSF) de acordo com a sua classificação.

2.3.1 Técnicas Globais de Redução de Dados em RSSF (Topológicas)

As técnicas globais de redução de dados em RSSF, também chamadas de *In-Network Processing Techniques* são caracterizadas por realizar a redução de dados através do processamento aplicado pelos nós intermediários da rede após o envio das

mensagens. Neste cenário, uma das técnicas amplamente encontrada na literatura é a Agregação de Dados.

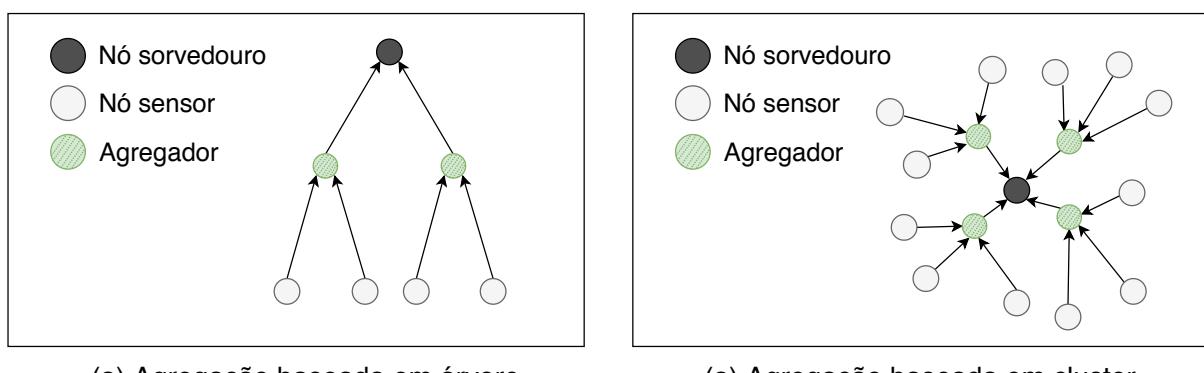
2.3.1.1 Agregação de Dados

Agregação de dados é uma técnica bastante utilizada em Redes de Sensores sem Fio com o objetivo principal de agrregar dados provenientes de diferentes nós. Nessa abordagem, os nós definidos como agregadores são responsáveis por coletar os dados de diferentes sensores e enviá-los de uma só vez, ao invés de simplesmente retransmiti-los. A consequência do uso desta técnica é a redução no numero de pacotes transmitidos para a rede, bem como, maior economia de energia (DHAND; TYAGI, 2016).

Segundo Raut et al. (2018), a topologia da rede de sensores deve ser levada em consideração para que o processo de agregação de dados seja realizado com maior eficácia possível. Consequentemente, as abordagens para agregação de dados possuem características específicas de acordo com a forma em que os sensores estão dispostos.

Os mecanismos de agregação de dados podem ser divididos de acordo com as seguintes classificações: agregação sem estrutura, agregação baseada em estrutura, e agregação baseada em estrutura híbrida (SPANDAN et al., 2013). O mecanismo de agregação sem estrutura geralmente é utilizado em redes nas quais os nós sensores são implantados de forma aleatória, formando uma topologia plana⁴ (MAMUN, 2012). O mecanismo de agregação de dados baseado em estrutura é aplicado com base em duas diferentes abordagens. São elas: abordagens baseadas em árvore (DAFLAPURKAR; GANDHI; PATIL, 2017) e abordagens baseadas em *cluster* (GILBERT et al., 2018), como ilustrado na Figura 2.3.

Figura 2.3 – Abordagens de Agregação em RSSF: (a) Árvore; (b) Cluster



⁴ Do Inglês: *Flat Topology*

Nas abordagens de agregação de dados baseadas em árvore, como ilustrado na Figura 2.3(a), todos os sensores estão dispostos na forma de uma árvore, na qual o processo de agregação acontece de forma hierárquica, sendo que nós pais agregam dados de nós filhos (DAGAR; MAHAJAN, 2013). Nas abordagens baseadas em *cluster*, como ilustrado na Figura 2.3(b), a rede inteira é disposta na forma de vários *clusters*, sendo que cada *cluster* é formado por vários sensores. Nesse contexto, é necessária a escolha de um nó principal, também chamado de *Cluster Head*, que irá desempenhar a função de agregador (SIRSIKAR; ANAVATTI, 2015).

O mecanismo de agregação de dados baseado em estrutura híbrida, por sua vez, é utilizado apenas em redes cuja estrutura é formada por duas ou mais topologias, por exemplo, árvore e *cluster*.

Alguns benefícios obtidos através do uso de técnicas de agregação de dados são listados por Pourghbleh e Navimipour (2017). São eles: melhorias na eficiência e acurácia da informação em toda a rede; diminuição das informações desnecessárias uma vez que existe redundância nos dados coletados pelos nós sensores; diminuição da carga no tráfego de dados e uso mais eficiente das fontes de energia dos nós. Por outro lado, o uso da técnica de agregação de dados exige que alguns fatores sejam analisados com maior rigor, como acurácia da informação em caso de falhas, latência, escalabilidade, e segurança.

- Acurácia: A acurácia da informação pode ser comprometida em casos de falhas do nó agregador, ou em outro nó que repetidamente envie dados errôneos;
- Latência: Ao utilizar uma técnica de agregação de dados, o nó agregador precisa esperar até que uma determinada quantidade de dados de outros nós seja recebida. Consequentemente, o agregador pode adicionar um determinado atraso no tempo de entrega das mensagens ao sorvedouro;
- Escalabilidade: Para que um mecanismo de agregação de dados em RSSF seja escalável, é necessário que o alto número de nós não interfira na performance das atividades de agregação (SPANDAN et al., 2013);
- Segurança: Se um nó sensor for comprometido, é possível que dados falsos sejam fornecidos a fim de enganar o agregador para que o mesmo receba leituras erradas (DHASIAN; BALASUBRAMANIAN, 2013).

A agregação de dados é uma técnica que vem ganhando destaque nos últimos anos e tem se mostrado muito útil para a redução das transmissões de dados em Redes de Sensores sem Fio. Dessa forma, é possível obter um consumo mais eficiente dos recursos da rede, bem como o aumento do seu tempo de vida.

2.3.1.2 *Fusão de Dados*

Em determinadas Redes de Sensores sem Fio (RSSF), é possível que cada nó da rede seja equipado com múltiplos sensores. Tais dispositivos permitem a detecção de vários fenômenos em um determinado ambiente de forma unificada. De acordo com Abdalgawad (2011), a fusão de dados pode ser definida como a coleta sistemática de dados detectados a partir de múltiplos sensores que devem ser eventualmente transmitidos ao sorvedouro de forma unificada.

Zhai, Jing e Vladimirova (2014) destaca que as arquiteturas de fusão de dados podem ser classificadas como: arquiteturas centralizadas e arquiteturas distribuídas. A arquitetura centralizada é a arquitetura mais simples. Todos os dados detectados são transmitidos para o nó sorvedouro que por sua vez funde os dados coletados. Nesta arquitetura, o nó sorvedouro controla toda a rede. As vantagens dessa arquitetura incluem facilidade de implementação e a identificação célere caso hajam falhas nos nós da rede. Como desvantagem, têm-se o consumo desequilibrado de energia, que pode causar indisponibilidade dos nós em determinadas áreas (MHATRE; ROSENBERG, 2004).

Já a arquitetura descentralizada é mais robusta e amplamente utilizada em Redes de Sensores sem Fio (RSSF). Nesta arquitetura, a fusão dos dados não ocorre em um ponto fixo. Ao invés disso, cada nó pode desempenhar a fusão de dados baseado nas próprias detecções e também nos dados provenientes dos nós vizinhos. A arquitetura descentralizada provê alto nível de tolerância a falhas, além de gerar impacto positivo no consumo de energia, uma vez que a quantidade de transmissão de dados pode ser reduzida (ZHAI; JING; VLADIMIROVA, 2014).

Com relação às arquiteturas descentralizadas, Chakraborty e Das (2017) destacam que existem quatro principais tipos de fusão de dados. São eles: fusão de baixo nível; fusão de médio nível, fusão de alto nível e fusão multinível. Primeiramente, na fusão de baixo nível, dados brutos (dados de diferentes sensores no mesmo nó) são unificados uma vez que há uma determinada correlação entre eles. Dessa forma, ao invés de transmitir estes dados de forma isolada, eles são combinados, a fim de que sejam transmitidos de forma unificada (ZHANG; LAI; CHAO, 2014). Por isso, a técnica de fusão de dados também é classificada como uma técnica local (não-topológica).

Na fusão de médio nível, informações sobre recursos e outros detalhes de determinados nós da rede são fundidos a fim de obter um mapa de recursos que pode ser usado para outras tarefas. Também é conhecido como fusão de nível de recurso. A fusão de alto nível, por sua vez, é responsável por receber determinadas informações que são unificadas (fundidas), a fim de que representem maior embasamento para tomada de decisões a nível global (CHAKRABORTY; DAS, 2017).

Por fim, na fusão multinível, a fusão de dados é realizada com o objetivo de eliminar a transmissão redundante de dados, reduzindo a quantidade total de transmissões de dados na rede. Especificamente, esta técnica é baseada na agregação de dados (abordada na subseção anterior) provenientes de diversos nós sensores vizinhos (ZECHINELLI-MARTINI; BUCCIOL; VARGAS-SOLAR, 2011). Neste caso, quando dois ou mais nós sensores enviam os mesmos dados (dados redundantes) para um nó em particular onde ocorre a agregação, então ele encaminha apenas uma única cópia desse dado ao sorvedouro, reduzindo o tráfego na rede.

Com base no exposto, entende-se que a fusão de dados reduz a quantidade de transmissão de dados em RSSF através da junção de dados coletados por diferentes sensores em um único dispositivo, beneficiando-se da correlação espacial e temporal entre os mesmos. Outra forma de reduzir a quantidade de transmissões em RSSF é a realização da agregação de dados provenientes de diferentes nós, com a finalidade de reduzir a redundância. Existem ainda outras abordagens, como o trabalho de Manjunatha, Verma e Srividya (2008), que realiza a fusão de dados provenientes de diferentes sensores através de métodos baseados em regras difusas.

2.3.2 Técnicas Locais de Redução de Dados em RSSF (Não-Topológicas)

As técnicas locais (não-topológicas) de redução de dados em RSSF são baseadas na redução de dados realizada de maneira local (nos sensores), antes que as mensagens sejam enviadas para a rede. A Fusão de Dados de baixo nível é um exemplo de técnica local, uma vez que envia dados de forma unificada a fim de reduzir o número de transmissões realizada pelos nós sensores. Outras técnicas podem ser destacadas, como Predição de Dados, *Compressive Sensing* e Análise da Dispersão.

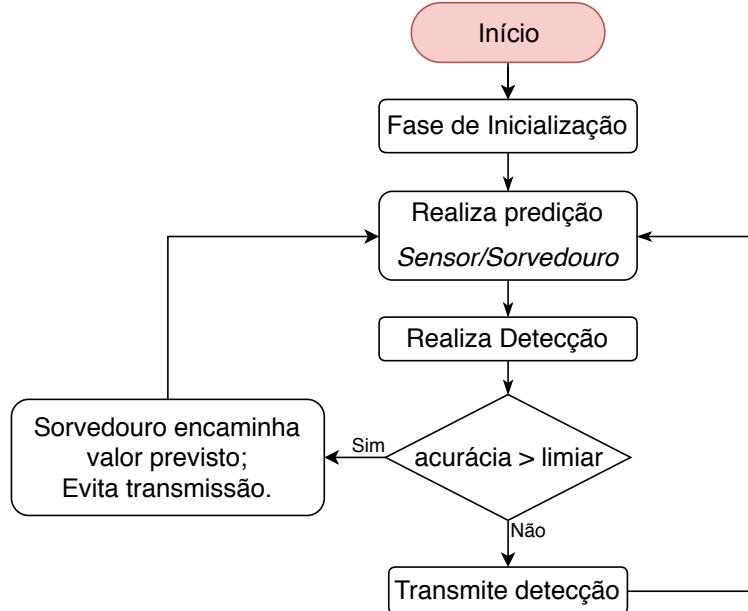
2.3.2.1 *Predição de Dados*

A predição de dados é uma técnica que visa estimar valores futuros com base em dados históricos. Baseado neste princípio, Soua e Minet (2011) afirmam que o histórico das detecções de um determinado nó sensor pode ser utilizado para que um nó sorvedouro realize a predição de detecções futuras. O objetivo principal é evitar que o nó sensor envie um dado que já foi previsto no sorvedouro, levando em consideração um certo nível de acurácia da predição. Este é um exemplo de um esquema de predição única, no qual a predição ocorre apenas no sorvedouro.

Existem na literatura, abordagens com esquemas de predição dupla. Nesses casos, a predição ocorre tanto no sensor quanto no sorvedouro. Nestas abordagens, são utilizados modelos de predição determinísticos, para que o mesmo resultado seja obtido tanto no sensor quanto no sorvedouro (DIAS; BELLALTA; OECHSNER, 2016a). Dessa forma, o nó sensor pode verificar a acurácia da predição localmente e decidir

quando uma transmissão não deve ser realizada. A Figura 2.4 representa o fluxo de execução do esquema de predição dupla.

Figura 2.4 – Fluxo de Execução do Esquema de Predição Dupla. Fonte: Próprio Autor



No esquema de predição dupla, sempre que o nó sensor detecta um valor que difere consideravelmente do valor previsto (baixa acurácia), o valor real detectado é enviado para o sorvedouro, que por sua vez, substitui o valor previsto pelo valor real. Consequentemente, os nós sensores podem diminuir significativamente a quantidade de transmissões de dados.

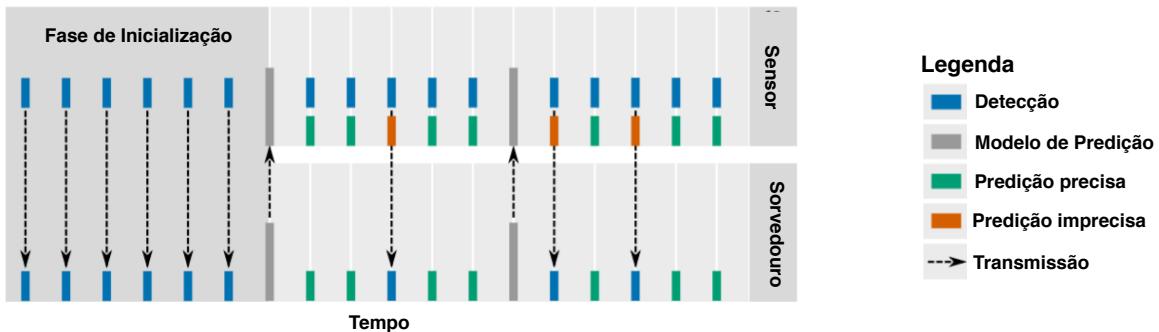
A fase de inicialização expressa na Figura 2.4 comprehende a coleção das primeiras detecções realizadas por um determinado nó sensor. Elas servirão de base para que o algoritmo de predição possa realizar as previsões futuras com o maior nível de acurácia possível. No caso dos trabalhos (FATHY; BARNAGHI; TAFAZOLLI, 2018; SANTINI; ROMER, 2006), as 10 primeiras detecções dos nós são utilizadas na fase de inicialização.

De acordo com Borgne, Santini e Bontempi (2007), alguns modelos de predição dependem de parâmetros cuja a identificação dos seus valores não é uma tarefa trivial, especialmente quando nenhum conhecimento a-priori está disponível. Essa dificuldade cresce à medida em que aumenta o número de parâmetros que um modelo possui.

Os esquemas de predição dupla requerem que sensor e sorvedouro executem o mesmo modelo de predição. Além disso, todos os valores de parâmetros precisam ser compartilhados entre ambos dispositivos. Por isso, a escolha do modelo de predição é um passo fundamental neste processo. Na Figura 2.5, é ilustrado um esquema

de predição dupla no qual o sorvedouro escolhe um determinado modelo (algoritmo) de predição e então notifica o nó sensor sobre a sua escolha.

Figura 2.5 – Esquema de Predição Dupla: Compartilhamento do Algoritmo de Predição. Adaptado de (DIAS; BELLALTA; OECHSNER, 2016a)



No exemplo expresso na Figura 2.5, o esquema de predição dupla contém uma fase de inicialização, na qual todas as detecções do nó sensor são enviadas ao nó sorvedouro sem a aplicação de qualquer técnica de redução de dados. Nessa fase, as detecções formam uma coleção inicial de dados para que sirvam de base para a previsão de detecções futuras. Após a fase de inicialização, é realizada a etapa de escolha do modelo de predição. Tal escolha é realizada pelo nó sorvedouro, que notifica o nó sensor sobre o modelo de predição escolhido, pois ambos precisam executar o mesmo modelo. No decorrer do funcionamento da rede, o sorvedouro pode optar pela troca do modelo de predição, visando obter previsões mais assertivas.

Realizar a escolha do modelo de predição no sorvedouro, explora a vantagem do maior poder computacional e menor restrição de consumo de energia que geralmente os sorvedouros possuem. Ainda assim, outras abordagens podem ser encontradas na literatura, como atribuir ao nó sensor a tarefa de escolher o modelo de predição (KHO; ROGERS; JENNINGS, 2009) ou permitir que cada dispositivo decida por um modelo de predição diferente (SANTINI; ROMER, 2006).

2.3.2.2 Compressive Sensing (CS)

Em busca da minimização da sobrecarga na comunicação em Redes de Sensores sem Fio, muitos pesquisadores têm explorado o ramo do processamento de sinais na coleta de dados em RSSF (POOR, 2006; BAL; SHEN; GHENNIWA, 2009). Nesse contexto, a redução de dados acontece através da possibilidade de reconstruir um sinal a partir de uma pequena amostragem (QAISAR et al., 2013). Um exemplo disso é o teorema da amostragem de Nyquist–Shannon, também chamado de teorema da amostragem. Este teorema afirma que é necessária uma taxa de amostragem de no mínimo o dobro da frequência máxima presente no sinal, para recuperar

o sinal sem serrilhamento (*aliasing*) nem perda de informação (MATUSIAK; ELDAR, 2012).

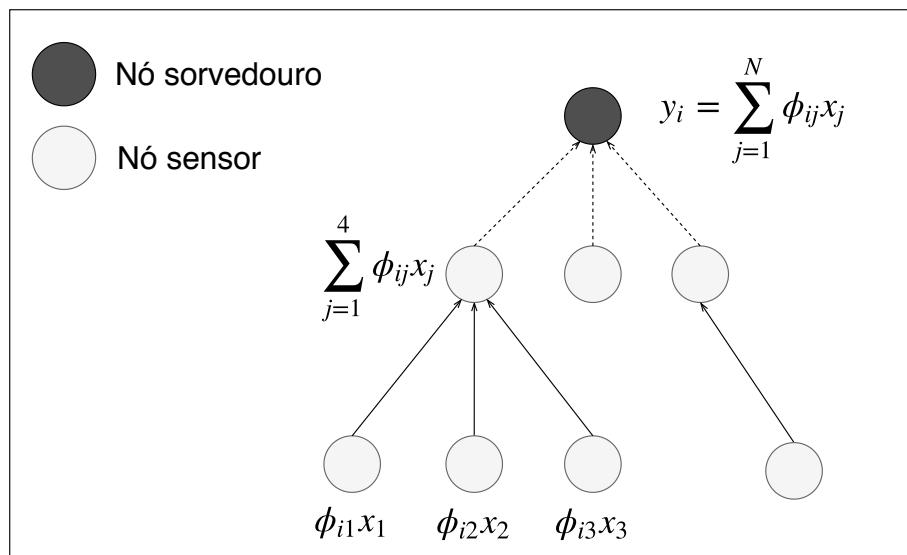
Uma das técnicas derivadas do processamento de sinais que ganhou bastante atenção nos últimos anos é a técnica *Compressive Sensing* (CS)⁵. De acordo com Ji e Carin (2007), através do uso da técnica de CS é possível recuperar certos sinais com uma quantidade de amostras muito menor que as requeridas pelo teorema de Nyquist–Shannon. Para que esta técnica possa ser aplicada, o sinal a ser reconstruído deve conter duas propriedades básicas: esparsidade e independência linear (CANDÈS; WAKIN, 2008).

Para reduzir a dimensionalidade de x , a técnica de CS aplica uma matriz de detecções $\Phi \in \mathbb{R}^{m \times n}$ em x a fim de obter um sinal m -dimensional $y \in \mathbb{R}^m$. Segundo Zheng et al. (2017), um sinal k -esparso pode ser precisamente recuperado com alta probabilidade através da propriedade *Restricted Isometry Property* (RIP)⁶, propriedade que estabelece uma regra na qual as linhas da matriz Φ não devem representar esparsamente as colunas de x . A propriedade RIP garante a recuperação de x a partir do sinal comprimido y se $m \geq k \log n/k$ (CANDÈS; ROMBERG; TAO, 2006).

A reconstrução do sinal x a partir de y é considerado um problema NP-Difícil. Porém, Masoum, Meratnia e Havinga (2013) destacam que a reconstrução pode ser realizada através da resolução do problema da minimização- l_1 .

Na Figura 2.6 é ilustrado o funcionamento da técnica *Compressive Sensing* em uma Rede de Sensores sem Fio.

Figura 2.6 – Utilização da Técnica de CS em RSSF. Adaptado de (XIE; JIA, 2014)



⁵ Em Português: Detecção Compressiva

⁶ Em Português: Propriedade de Isometria Restrita

Como observado na Figura 2.6, a técnica de CS pode ser aplicada em RSSF a fim de reduzir a sobrecarga na rede, bem como aumentar a quantidade de energia dos nós sensores. Neste exemplo, considera-se uma rede que possui apenas um nó sorvedouro e N nós sensores. O vetor x será representado pelas detecções dos sensores, contendo N elementos, sendo um para cada sensor. Dessa forma, quando o valor de k é pequeno, ao invés de transmitir N dados ao sorvedouro, é possível enviar um número menor de projeções de x ao sorvedouro (XIE; JIA, 2014).

No nó sorvedouro, após coletar o vetor y , os dados originais x podem ser recuperados através da execução do algoritmo de minimização- l_1 . Outras heurísticas também podem ser utilizadas para esse fim, como *Orthogonal Matching Pursuit* (OMP) (TROPP; GILBERT, 2007).

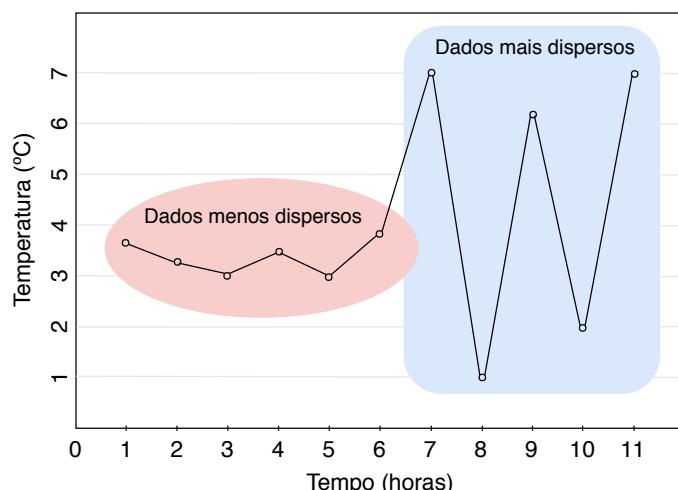
2.3.2.3 Análise da Dispersão

A técnica de redução de dados baseada na análise da dispersão consiste em analisar a variabilidade dos valores detectados por um determinado nó sensor. De acordo com VC, K e Kumar (2014), a análise da dispersão pode ser utilizada para evitar a transmissão de dados redundantes, ou seja, detecções consecutivas que não demonstram variabilidade.

Para analisar a variabilidade dos valores detectados, é necessário que seja definida uma medida de dispersão. Segundo Natrella (2010), a medida de dispersão deve ser representada por um número real não-negativo que é zero quando os dados são iguais. Esse número aumenta à medida em que os dados se divergem.

Na Figura 2.7 é ilustrado um exemplo de dispersão em dados de temperatura coletados por um nó sensor, no qual os seis primeiros valores são menos dispersos que os demais.

Figura 2.7 – Exemplo de Dispersão nos Dados. Fonte: Próprio Autor



Uma das formas de aferir a dispersão entre as detecções é calcular a diferença obtida na subtração entre os valores absolutos de duas detecções consecutivas. Esta abordagem é utilizada no trabalho de Almhana, Choulakian e Almhana (2017), que realiza a análise da dispersão para reduzir o número de transmissões de dados em Redes de Sensores sem Fio (RSSF).

Além de ser utilizada para evitar a transmissão de dados quando não há mudança no ambiente sensoreado, a técnica de análise da dispersão pode ser aplicada para que as transmissões de dados só sejam realizadas quando a mudança no ambiente sensoreado for significativa. Para tal, é necessária a definição de um limiar de dispersão a fim de que detecções pouco dispersas não sejam transmitidas.

Uma vez que a transmissão de valores pouco dispersos é suprimida, é necessária a utilização de um método de reconstrução de dados para que, no nó sorvedouro, novos dados sejam gerados a fim de que substituam aqueles que não foram transmitidos pelos nós sensores. Uma das formas de realizar a substituição dos valores não transmitidos é utilizar a média dos mesmos. Nesse caso, o algoritmo de reconstrução de dados repete a média dos valores pouco dispersos de acordo com a quantidade de transmissões que não foram realizadas por conta da baixa dispersão. Neste processo, são introduzidos erros, uma vez que os dados reconstruídos podem se distanciar dos dados reais (OLIVEIRA; KNIESS, 2019).

Outra possibilidade é realizar a reconstrução de dados baseando-se na tendência da série temporal do conjunto de valores pouco dispersos. Neste caso, o processo de reconstrução utiliza como base a primeira e última detecção do conjunto, visando gerar valores uniformemente espaçados entre eles, formando uma linha de tendência. Dessa forma é possível alcançar maior aproximação dos dados reais, diminuindo o erro nos dados reconstruídos (OLIVEIRA; KNIESS; MARQUES, 2019).

Dada a característica de realizar a redução de dados no nó sensor, a técnica de redução de dados baseada na análise da dispersão se assemelha às técnicas de redução de dados baseadas em predição de dados e em *Compressive Sensing* (técnicas locais). No entanto, as técnicas de redução de dados baseadas em predição e em *Compressive Sensing* podem exigir alto poder computacional dos nós, principalmente dos sorvedouros. O método baseado na análise da dispersão, por sua vez, é beneficiada pela baixa dispersão nos dados detectados, alcançando tal redução de forma mais simples, sem realizar um alto grau de computação, e consequentemente, podendo ser aplicada facilmente em dispositivos com maiores restrições.

Na próxima seção, serão abordados conceitos sobre a predição de séries temporais, bem como conceitos sobre modelos e algoritmos de predição, baseados em estatística e aprendizado de máquina.

2.4 PREDIÇÃO DE SÉRIES TEMPORAIS

As aplicações que utilizam sensores para fins de monitoramento, geralmente dão origem a coleções de dados ordenados pelo tempo, no formato de séries temporais. Nesses casos, é comum que informações históricas sejam utilizadas para realizar a previsão de valores futuros. Ao utilizar algoritmos baseados em modelos de predição de séries temporais, a informação histórica de uma determinada série temporal é processada a fim de prever valores ou tendências futuras.

De acordo com Boukary (2016), os modelos de predição de séries temporais podem ser classificados em três: Modelos baseados em estatística, modelos baseados em aprendizado de máquina, e modelos híbridos, cuja a implementação mescla conceitos de estatística e aprendizado de máquina.

Nas subseções a seguir, serão apresentados modelos de predição de séries temporais do estado-da-arte encontrados na literatura. Dentre os modelos estatísticos, destaca-se o modelo *Auto-Regressive Integrated Moving Average* (ARIMA) (BOX et al., 2015), apresentado na Subseção 2.4.1. Dentre os modelos baseados em *Machine Learning*, destacam-se os seguintes modelos: *Support Vector Machine* (SVM)⁷ (SAPANKEVYCH; SANKAR, 2009), apresentado na Subseção 2.4.2, e Redes Neurais Artificiais (HE; XU, 2010), apresentado na Subseção 2.4.3.

2.4.1 *Auto-Regressive Integrated Moving Average* (ARIMA)

O modelo ARIMA é um modelo estatístico de previsão de séries temporais que visa encontrar auto-correlações nas observações de uma determinada série temporal a fim de realizar previsões futuras.

De acordo com Marcellino, Stock e Watson (2006), o modelo ARIMA é composto por dois modelos: auto-regressivo (AR) e médias móveis (MA). O modelo auto-regressivo (AR) prevê a variável de interesse (y) utilizando uma combinação linear dos seus valores passados ($y_{t-1}, y_{t-2}, \dots, y_{t-p}$). A Equação 2.1 (MARCELLINO; STOCK; WATSON, 2006) representa um modelo auto-regressivo (AR) de ordem p .

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.1)$$

Na Equação 2.1, ε_t representa o ruído branco (sinal aleatório com igual intensidade em diferentes frequências) encontrado na série temporal, enquanto ϕ representa o parâmetro do modelo. A ordem p do modelo determina o número de valores históricos utilizados para prever o valor futuro.

⁷ Em Português: Máquina de Vetor de Suporte

Boukary (2016) destaca que no modelo de médias móveis (MA), por sua vez, são utilizados os erros das previsões passadas em um modelo baseado em regressão, ao invés dos valores históricos da variável de interesse como no modelo auto-regressivo (AR). Dessa forma, a Equação 2.2 (BOUKARY, 2016) representa um modelo de médias móveis (MA) de ordem q .

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (2.2)$$

Na Equação 2.2, ε_t também representa o ruído branco e θ representa o parâmetro do modelo.

Os modelos auto-regressivos (AR) e médias móveis (MA) lidam apenas com séries temporais estacionárias (DEB et al., 2017). Uma série temporal é estacionária se a média da série e a covariância entre as suas observações não mudam ao passar do tempo e não seguem tendências. Logo, séries temporais com tendência ou sazonalidade, não podem ser consideradas estacionárias (FARHATH; ARPUTHAMARY; AROCKIAM, 2016).

Para que a previsão de séries temporais não-estacionárias seja realizada, é necessária a aplicação de um processo chamado de diferenciação (HYNDMAN; ATHANASOPOULOS, 2018). A diferenciação objetiva transformar séries temporais não-estacionárias em séries temporais estacionárias através do cálculo da diferença entre os valores consecutivos da série temporal. O processo de diferenciação visa estabilizar a média da série temporal através da remoção de grandes variações, tendo como consequência a redução de comportamentos de tendência e sazonalidade.

O modelo ARIMA é um modelo de predição de séries temporais composto pela junção dos modelos auto-regressivo (AR) e médias móveis (MA), juntamente com o processo de diferenciação. O modelo ARIMA pode ser representado como expresso na Equação 2.3 (HYNDMAN; ATHANASOPOULOS, 2018).

$$y'_t = \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.3)$$

Na Equação 2.3, os parâmetros p , d e q são inteiros não negativos que se referem ao modelo auto-regressivo (AR), grau de diferenciação e modelo de médias móveis (MA) respectivamente.

2.4.2 **Support Vector Machine (SVM)**

Support Vector Machine (SVM) é uma técnica de aprendizado de máquina amplamente utilizada em diversos domínios, como na categorização de textos, análise de imagens e em Bioinformática (LORENA; CARVALHO, 2007). O SVM possui

um conjunto de métodos de aprendizado supervisionado para análise de dados e reconhecimento de padrões, além de poder ser aplicado tanto para classificação, como para regressão.

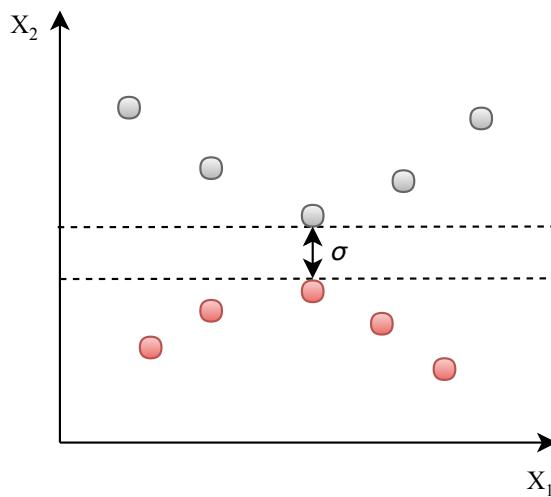
Em classificação, o SVM busca encontrar uma linha de separação entre pontos de classes diferentes dispostos em um determinado espaço. Segundo Albuquerque (2014), o objetivo principal é encontrar uma função de decisão na forma:

$$f(x) = \text{sinal}(w^t x - \gamma), \quad (2.4)$$

na qual x é um vetor de dimensão $p \times 1$ representando o valor de observações arbitrárias com p variáveis, w é o vetor de parâmetros de dimensão $p \times 1$ e γ é um parâmetro escalar denominado termo de viés.

A estimativa por máxima margem consiste em determinar w e γ de modo que as observações de classes diferentes fiquem o mais afastado possível umas das outras. Na Figura 2.8 é representado um exemplo em que os dados são separados através de dois segmentos de retas e σ representa a máxima margem que separaria os dois conjuntos de dados.

Figura 2.8 – Separação por Máxima Margem Linear. Fonte: Adaptado de (ALBUQUERQUE, 2014)



Existe uma derivação do SVM para o contexto de regressão e previsão de séries temporais, denominada *Support Vector Regression* (SVR), proposto por Drucker et al. (1997). O SVR é considerada uma técnica não-paramétrica, pois depende da utilização de núcleos, também chamados de *kernels*, que são funções matemáticas utilizadas para receber dados de entrada e transformá-los na forma requerida.

Para exemplificar, os dados sequenciais que compõe a entrada do algoritmo SVR podem ser compreendidos da seguinte maneira: $\{(x_1, y_1) \dots (x_l, y_l)\} \subset \mathbb{N} \times \mathbb{R}$, onde

\mathbb{N} denota o espaço dos padrões de entrada (SAPANKEVYCH; SANKAR, 2009). O SVR objetiva encontrar uma função $f(x)$ que tenha no máximo um determinado desvio (ϵ) dos valores reais obtidos em y_i nos dados de entrada. A Equação 2.5 (SAPANKEVYCH; SANKAR, 2009) define a função $f(x)$ para regressão linear:

$$f(x) = (w \times x) + b, \text{ sendo que } w \in \mathbb{N}, b \in \mathbb{R} \quad (2.5)$$

Smola e Schölkopf (2004) destacam que as funções para regressão buscam encontrar planura⁸ dos pesos (w) que pode ser medida pela norma euclidiana, i.e. $\|w^2\|$. Este problema pode ser expresso formalmente como um problema de otimização convexa, como na Equação 2.6 (SMOLA; SCHÖLKOPF, 2004):

$$\begin{aligned} & \text{minimizar} \quad \frac{1}{2} \|w^2\| \\ & \text{sujeito a} \quad \begin{cases} y_i - (w \times x_i) - b \leq \epsilon \\ (w \times x_i) + b - y_i \leq \epsilon \end{cases} \end{aligned} \quad (2.6)$$

É possível que não exista tal função $f(x)$ que satisfaça todas as restrições. Para lidar com estes casos, pode-se utilizar variáveis de folga ξ_i e ξ_i^* . Esta abordagem permite a ocorrência erros de regressão até o valor definido para ξ_i e ξ_i^* , e ainda assim satisfazem as condições exigidas (BASAK; PAL; PATRANABIS, 2007).

Ao incluir as variáveis de folga, obtém-se a fórmula primal, proposta inicialmente por Vapnik (1995):

$$\begin{aligned} & \text{minimizar} \quad \frac{1}{2} \|w^2\| + C \sum_{i=1}^N \xi_i + \xi_i^* \\ & \text{sujeito a} \quad \begin{cases} y_i - (w \times x_i) - b \leq \epsilon + \xi_i \\ (w \times x_i) + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (2.7)$$

A constante $C > 0$ representa o conflito de escolha entre a planura de $f(x)$ e a quantidade em que valores maiores que ϵ são tolerados. Nesse caso, lida-se com função de perda ϵ -insensível $|\xi|_\epsilon$ descrita na Equação 2.8 (SMOLA; SCHÖLKOPF, 2004):

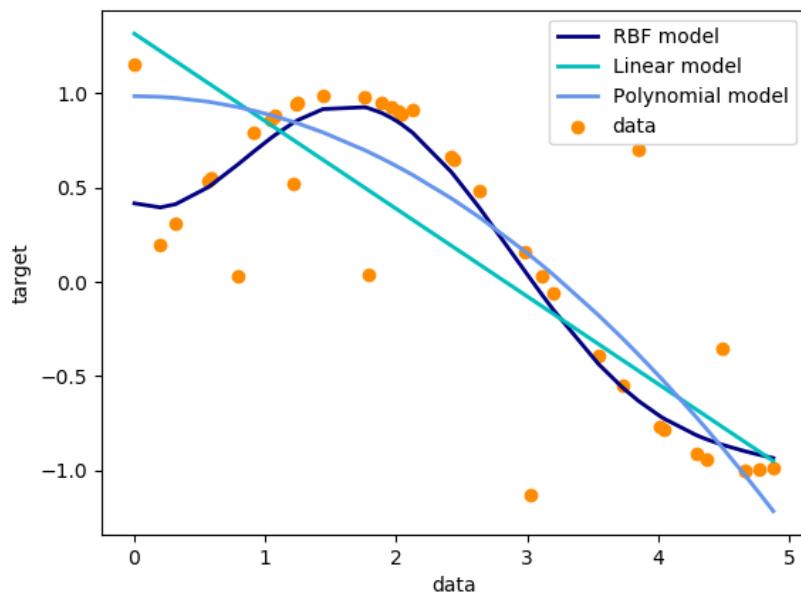
$$|\xi|_\epsilon = \begin{cases} 0 & \text{se } |\xi| < \epsilon \\ |\xi| - \epsilon & \text{se } |\xi| \geq \epsilon \end{cases} \quad (2.8)$$

⁸ Do Inglês: *Flatness*

De acordo com Rüping (2001), Rubio et al. (2007), vários *kernel*s podem ser utilizados em SVR, como: kernel linear, kernel função de base radial⁹, kernel quadrado mínimo¹⁰, dentre outros.

No caso da previsão de valores futuros, destaca-se o *kernel* de função de base radial pelo fato de se ajustar melhor às amostras de dados em problemas de regressão não-linear, como ilustrado na Figura 2.9.

Figura 2.9 – Funções de *kernel* lineares e não-lineares do modelo SVR. Fonte: (SCIKIT-LEARN, 2018)



2.4.3 Redes Neurais Artificiais

Redes Neurais Artificiais são modelos computacionais inspirados em redes neurais biológicas, que constituem os cérebros animais. Tais modelos buscam aprender a realizar determinadas tarefas através da análise de exemplos. As Redes Neurais Artificiais têm como principal objetivo aprender a resolver problemas complexos em uma quantidade razoável de tempo (GERVEN; BOHTE, 2018).

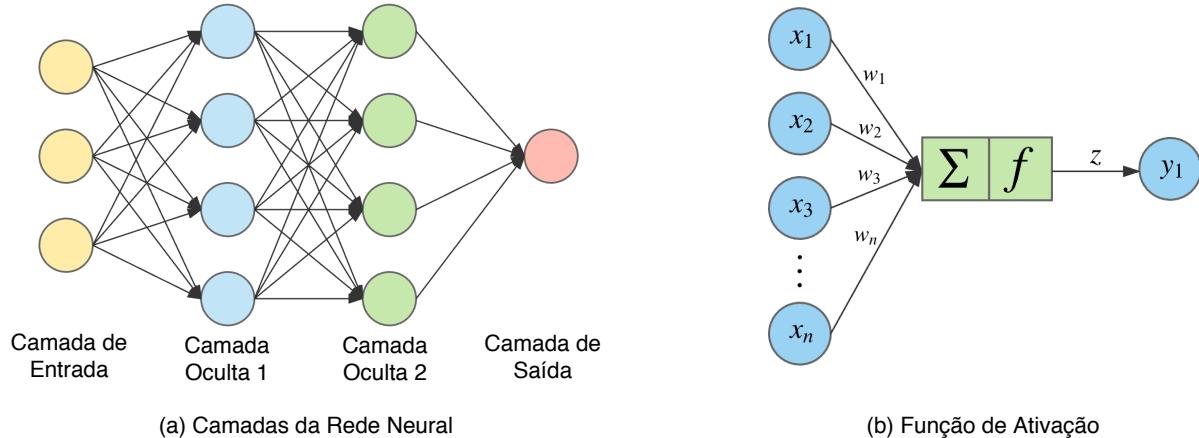
Atualmente, Redes Neurais Artificiais têm sido utilizadas massivamente em diversas áreas de aplicação, como classificação de imagens (FADAEDDINI; ESHGHI; MAJIDI, 2018), reconhecimento de fala (LOKESH et al., 2018), processamento de linguagem natural (GOLDBERG, 2017) e predição de séries temporais (LANG et al., 2018).

⁹ Do Inglês: *Radial Basis Function* (RBF)

¹⁰ Do Inglês: *Least Square* (LS)

As Redes Neurais Artificiais são compostas por uma camada de entrada, uma ou mais camadas ocultas, e uma camada de saída. Cada camada contém um determinado número de neurônios, sendo que cada neurônio é conectado a todos os neurônios da camada seguinte através de sinapses. Na Figura 2.10(a) a disposição dos neurônios em uma Rede Neural Artificial é ilustrada.

Figura 2.10 – Rede Neural Artificial. Fonte: Adaptado de (DERTAT, 2017)



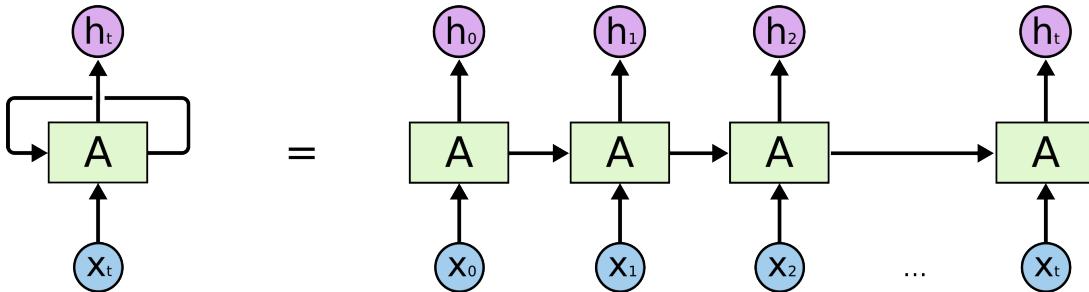
De acordo com Dertat (2017), a função das sinapses é atribuir um peso aos dados por ela transferidos. Os neurônios da camada de entrada são responsáveis pela inserção dos dados na rede, enquanto os neurônios das camadas ocultas recebem as somas ponderadas das suas entradas e repassam novos dados gerados através de uma função de ativação não linear, como ilustrado na Figura 2.10(b).

Uma vez que a conexão entre os neurônios representados na Figura 2.10 não formam ciclos, essas redes são chamadas de Redes Neurais *Feedforward*. Contudo, existem redes neurais em que as conexões entre seus neurônios formam ciclos, com a finalidade de capturar informações temporais e contextuais em dados de séries temporais. Essas redes são chamadas de Redes Neurais Recorrentes (LIU; SULLIVAN, 2018).

Redes Neurais Recorrentes têm a capacidade de receber dados da camada de entrada e mantê-los para serem utilizados na próxima iteração. Desta forma, o fluxo da informação não segue em um único sentido, nem o resultado final depende exclusivamente da entrada atual, mas também das entradas anteriores.

Goodfellow, Bengio e Courville (2016) explica que o estado oculto da Rede Neural Recorrente recebe informações das variáveis independentes, assim como de seu próprio resultado de processamento da iteração anterior, como ilustrado na Figura 2.11.

Figura 2.11 – Neurônio da Camada Oculta em uma RNN. Fonte: (OLAH, 2014)



Neste exemplo, X_t é representado pelo dado de entrada na iteração de tempo t , o qual será processado pelo neurônio A . h_t é representado pela saída do neurônio A na iteração de tempo t , que pode ser utilizada nas próximas iterações pelo mesmo neurônio, como comportamento padrão de recorrência.

Goodfellow, Bengio e Courville (2016) destaca ainda que a atividade dos neurônios em Redes Neurais Recorrentes é similar ao comportamento de uma memória de curto prazo, sendo bastante útil no contexto de predição de séries temporais, tendo em vista que os dados das camadas de entrada e saída são dados sequenciais.

Nesse sentido, cada amostra dos dados precisa ser uma sequência, cujo formato é composto por períodos de tempo e variáveis. Assim, quando várias amostras são empilhadas para montar a base de dados, elas terão um formato 3D: número de amostras, períodos de tempo e variáveis. Dessa forma, os dados poderão ser submetidos a *sets* (conjuntos) de treino e teste, sendo que cada amostra deve conter a defasagem de uma ou mais linhas para que os dados mantenham o padrão sequencial.

Como o objetivo do modelo se trata de previsão, não é ideal simplesmente separar aleatoriamente os dados para os conjuntos de treino e teste. Para que a análise seja uma boa aproximação do cenário real, é preciso separar os últimos dados da série de tempo para servirem de conjunto de teste, por exemplo, os últimos 25% dos dados.

2.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram apresentados conceitos sobre Internet das Coisas, Redes de Sensores sem Fio, técnicas para redução na transmissão de dados em RSSF e modelos de previsão de séries temporais.

As técnicas apresentadas para redução na transmissão de dados em RSSF foram as seguintes: Redução de dados baseada em previsões, Agregação de dados, Fusão de dados, *Compressive Sensing* (CS) e Análise da dispersão. As técnicas

apresentadas para previsão de séries temporais foram as seguintes: *Auto-regressive Integrated Moving Average* (ARIMA), *Support Vector Machine* (SVM) e Redes Neurais Artificiais.

A abordagem proposta neste trabalho baseia-se na análise da dispersão dos dados coletados pelos nós sensores, visando efetuar transmissões de dados apenas quando há mudanças significativas no ambiente sensoreado. Além disso, um processo de reconstrução é utilizado a fim de preencher os valores que deixaram de ser transmitidos pelo nó sensor.

Com base na análise da literatura, a abordagem proposta neste trabalho para redução na transmissão de dados a partir de nós sensores, descrita no Capítulo 4, se assemelha às técnicas de redução de dados baseada em predição e *Compressive Sensing* (CS), no sentido de realizar a redução de dados no sensor, antes dos dados serem transmitidos para a rede.

No próximo capítulo, trabalhos relacionados ao tema de redução de dados em RSSF são apresentados. Ao final do capítulo, apresenta-se uma comparação entre os trabalhos relacionados e a abordagem proposta.

3 TRABALHOS RELACIONADOS

Neste capítulo, alguns trabalhos que lidam com o tema da redução de dados em Redes de Sensores Sem Fio são apresentados. Os trabalhos foram divididos em seções de acordo com suas características. Trabalhos envolvendo implementações de métodos baseados em agregação de dados estão na Seção 3.1, enquanto trabalhos baseados em fusão de dados estão na Seção 3.2. Trabalhos sobre predição de dados são apresentados na Seção 3.3. Já os trabalhos sobre *Compressive Sensing* estão na Seção 3.4. Na Seção 3.5, apresenta-se um resumo comparativo dos trabalhos relacionados na Tabela 3.1. Por fim, na Seção 3.6 são apresentadas as conclusões do capítulo.

3.1 MÉTODOS BASEADOS EM AGREGAÇÃO DE DADOS

Xu et al. (2015) propuseram um esquema de agregação de dados, incorporando a técnica de detecção compressiva em uma hierarquia multinível para Redes de Sensores sem Fio de larga escala. Este esquema realiza a agregação hierárquica de dados com a finalidade de definir dinamicamente os níveis de compressão baseado nos diferentes tamanhos de *clusters* em vários níveis da árvore de agregação de dados. O principal objetivo desta abordagem é unir a detecção compressiva e agregação de dados para minimizar a sobrecarga da comunicação em Redes de Sensores sem Fio.

Ao invés de configurar apenas um nó como sorvedouro, é configurada uma hierarquia de *clusters* de vários níveis para a coleta de dados intermediários. A consequência disto é a redução do volume de dados na comunicação. As comparações mostram que essa redução é maior que em outras abordagens baseadas na detecção compressiva. A reconstrução dos dados é realizada por um algoritmo baseado em *Discrete Cosine Transform* (DCT)¹.

As simulações foram realizadas na plataforma de simulação SIDnet-SWANS (GHICA et al., 2008). Foi utilizado na simulação um *dataset* de temperatura da superfície terrestre do *Nation Data Buoy Center* (NDBC)², detectado a partir de 25 sensores localizados no Oceano Pacífico no mês de Abril de 2014 (XU et al., 2015). Também foi utilizado um *dataset* sintético, a fim de avaliar o desempenho de diferentes esquemas de agregação de dados baseados em detecção compressiva. A análise dos resultados demonstra que a abordagem proposta atinge alto nível de agregação de dados e maior

¹ Em Português: Transformada Discreta de Cosseno

² <http://www.ndbc.noaa.gov/>

redução do consumo total de energia da rede se comparado a outras abordagens do estado-da-arte.

No trabalho de Zegarra et al. (2016), é proposto o algoritmo chamado *Continuous Enhancement Routing Solution* (CER), que é uma abordagem centralizada que utiliza o Algoritmo Genético *Biased Random Key Genetic Algorithm* (BRKGA) para computar a árvore de roteamento da RSSF. O CER tem como objetivo principal encontrar árvores de roteamento próximas do ótimo, realizando a agregação de dados pelo maior tempo possível.

O fluxo de execução da solução proposta se dá da seguinte forma: Primeiramente, o algoritmo configura a rede realizando troca de mensagens até que o nó sorvedouro tenha informação suficiente sobre a rede como um todo. Em seguida, conforme os eventos aparecem e desaparecem, os nós sensores solicitam uma árvore de roteamento ao nó sorvedouro. O sorvedouro computa as árvores de roteamento de maneira contínua através do BRKGA. Na prática, ele continua alimentando a rede com as árvores de roteamento de forma contínua, desde que encontre árvores de roteamento aprimoradas. Dessa forma, o CER gera menos pacotes, consequentemente, maior taxa de agregação de dados.

Para a realização dos experimentos, o CER foi implementado no simulador SinalGo (MUSZNICKI; ZWIERZYKOWSKI, 2012) e foi comparado com os algoritmos *Shortest Path Tree* (SPT) (KRISHNAMACHARI; ESTRIN; WICKER, 2002), *Data Aggregation Aware Routing Protocol* (DAARP) (VILLAS et al., 2009) e *Dynamic Data Aggregation Aware Routing Protocol* (DDAARP) (VILLAS et al., 2010). Na análise de resultados, constatou-se que o CER gera maior sobrecarga quando comparado ao SPT. A alta quantidade de árvores de roteamento ocasiona em uma quantidade expressiva de pacotes que configuram a rede, porém, isto é compensado pela qualidade das árvores de roteamento, que levam a uma taxa mais elevada de agregação de dados. Constatou-se também que em grandes redes de sensores, a execução do BRKGA pode apresentar um alto custo computacional. Logo, essa solução pode não ser adequada para aplicações que exigem baixa latência.

O trabalho de Li et al. (2017) propõe um método balanceado de agregação de dados de longo prazo (*Lifetime Balanced Data Aggregation* - LBA). O objetivo principal do LBA é prolongar o tempo de vida da rede considerando sua dinamicidade e heterogeneidade, garantindo o mínimo de atraso na entrega de dados.

O LBA é controlado pelos parâmetros (ρ, Θ) definidos como um requisito de atraso fim-a-fim, em que cada dado proveniente dos nós sensores deve atingir o sorvedouro no tempo Θ após sua geração com uma probabilidade ρ mínima. Outra característica do LBA é a capacidade de ajustar os atrasos de agregação dos dados de

dispositivos vizinhos (ao longo do caminho sensor-sorvedouro) de maneira colaborativa, de modo que a vida útil entre eles possa ser balanceada sem aumentar o atraso fim-a-fim. Dessa forma, o requisito de atraso de fim-a-fim é satisfeito, balanceando a vida útil dos dispositivos vizinhos, e consequentemente, estendendo a vida útil da rede inteira. Essa característica torna o LBA um algoritmo distribuído e escalável, uma vez que ocorre apenas entre vizinhos.

Os experimentos foram realizados através de uma rede *testbed* de 32 nós TelosB, formando uma topologia em árvore com apenas um nó sorvedouro. Os nós foram configurados para aguardar um determinado atraso de agregação, unir todos os pacotes recebidos ou criados durante esse tempo de espera em um único pacote para então encaminhar esse pacote para o próximo nó até chegar ao sorvedouro.

Os resultados demonstram que com o uso do LBA, a rede atinge maior tempo de vida se comparado a outras técnicas encontradas na literatura. Os benefícios são ainda mais interessantes quando todos os nós da rede iniciam atividade com a mesma quantidade de energia em suas baterias. Porém, mesmo nos testes onde a energia residual dos nós varia, o tempo de vida da rede é estendido. Uma fraqueza deste modelo é o fato de apenas considerar redes com um único nó sorvedouro.

3.2 MÉTODOS BASEADOS EM FUSÃO DE DADOS

No trabalho de Callegaro (2014) é apresentada uma proposta de arquitetura de fusão de dados para o monitoramento remoto de ambientes com Redes de Sensores sem Fio (RSSF). Nesse sentido, os processos propostos pela arquitetura são voltados para a diminuição da quantidade de transmissões de dados em nós sensores de baixo custo, bem como para a detecção de valores atípicos *outliers* que normalmente afetam a qualidade do monitoramento.

A arquitetura proposta leva em consideração uma RSSF configurada com uma topologia hierárquica, organizada na forma de *clusters* de sensores de pressão atmosférica. Além disso, a arquitetura é composta por duas camadas de fusão. São elas: camada de fusão local e camada de fusão de baixo nível. Na camada de fusão local, são realizadas fusões de dados gerados localmente no nó sensor, a fim de reduzir a quantidade de dados a ser trafegada para a rede. Na camada de fusão local, as fusões são realizadas através de uma modificação no algoritmo *Confidence-Weighted Averaging* (CWA) (DREDZE; CRAMMER; PEREIRA, 2008).

Na camada de fusão de baixo nível, por sua vez, a fusão de dados é realizada pelos nós intermediários, bem como pelo nó sorvedouro. Nesta camada também é realizada a detecção de valores anômalos (*outliers*) a fim de detectar se a qualidade do monitoramento está sendo afetada pela qualidade do nó sensor.

Os resultados obtidos pelo autor Callegaro (2014) mostram que a utilização da arquitetura proposta é capaz de reduzir significativamente a quantidade de transmissões de dados em RSSF, bem como identificar a existência de *outliers* nos dados coletados pelos sensores, a fim de não comprometer a qualidade do monitoramento realizado.

Já no trabalho de Zou e Liu (2015) foi proposto um modelo de detecção de eventos baseado na técnica de fusão de dados focada em Redes de Sensores sem Fio heterogêneas. Nesse sentido, uma aplicação de monitoramento de incêndios foi utilizada como um exemplo de detecção de eventos.

O modelo de detecção de evento proposto por Zou e Liu (2015) é baseado em fusão de dados e fusão de informações. No nível de fusão de dados, informações úteis são extraídas dos dados redundantes provenientes dos sensores. A partir de então, estes dados são tratados em um segundo nível de fusão, chamado fusão de informações. Neste segundo nível, as informações fundidas são utilizadas com a finalidade de detectar eventos específicos através de um algoritmo genético chamado *Data Fusion for Heterogeneous WSN Based*.

Por fim, os experimentos foram conduzidos em uma série de simulações através do simulador OMNeT++ (VARGA; HORNIG, 2008). Em comparação com outros métodos, o algoritmo de detecção de evento baseado em fusão de dados proposto pode efetivamente encontrar o evento através da detecção do estado de notificação e estado de alerta, alcançando resultados promissores em termos de qualidade e eficiência.

3.3 MÉTODOS BASEADOS EM PREDIÇÃO DE DADOS

Recentemente, a redução de dados em Redes de Sensores Sem Fio baseada na utilização de modelos de predição tem atraído grande interesse na comunidade científica, sendo amplamente discutida. Diversos autores publicaram pesquisas relacionadas ao desenvolvimento de modelos de predição para redução de dados em RSSF (YANG; WANG; ZHANG, 2008; SONI; CHAND; SINGH, 2012; TAN; WU, 2016; TAYEH et al., 2018; A..., 2019), além de pesquisas secundárias (*surveys*) sobre o assunto (ANASTASI et al., 2009; DIAS; BELLALTA; OECHSNER, 2016a; GANJEWAR et al., 2018).

No trabalho proposto por Dias, Bellalta e Oechsner (2016b), os autores adotam o esquema de dupla predição (no sensor e no sorvedouro) com a finalidade de prever o valor das detecções futuras. Dessa forma, quando a predição no sensor possui boa acurácia, o dado original não é enviado. Ao invés disso, o sorvedouro encaminha o valor resultante da predição, provendo assim menor sobrecarga na comunicação

interna da rede. O parâmetro (t_{min}) é utilizado para definir a acurácia mínima para que o dado original não seja enviado.

Neste trabalho, os métodos de predição de dados do estado-da-arte foram investigados com a finalidade de atestar a capacidade de manter a qualidade dos valores detectados pelas RSSF. Os algoritmos utilizados foram os que seguem: *Constant Predictions*, Método Linear, Método *Simple Mean*, ARIMA (*Autoregressive Integrated Moving Average*), e Redes Neurais Artificiais. Além disso, Dias, Bellalta e Oechsner (2016b) propõe um novo modelo de transmissão de dados em redes de sensores, visando combinar o esquema de dupla predição *Dual Prediction Scheme* (DPS) com técnicas de agregação de dados.

Nos experimentos, foram utilizados conjuntos de dados (*datasets*) de aplicações de monitoramento e aplicações de rastreamento. Por exemplo, utilizou-se os conjuntos de dados das aplicações de monitoramento que incluem dados de 54 sensores instalados no laboratório de pesquisa *Intel Berkeley Research Laboratory* entre Fevereiro e Abril de 2004 (MADDEN, 2004). Este *dataset* contém detecções de temperatura, umidade, luminosidade e consumo de energia e encontra-se disponível no sítio do laboratório³. Também foi utilizado o *dataset* do projeto Sensorscope (INGELREST et al., 2010), que são dados provenientes de uma rede de sensores distribuída em uma geleira na cidade de Lausanne - Suíça. Os conjuntos de dados de rastreamento são: dados do movimento de uma bola (sinteticamente criado para simular um objeto saltando no chão algumas vezes) e dados de corrida de rua, que consiste nas coordenadas de *Global Positioning System* (GPS) de uma pessoa enquanto corre pela cidade de Barcelona, na Espanha.

O modelo proposto por Dias, Bellalta e Oechsner (2016b) foi implementado no simulador Omnet++ (VARGA; HORNIG, 2008) com o propósito de realizar avaliações específicas para os diferentes valores de parâmetros (t_{min}). O DPS é composto por duas tarefas básicas que são executadas por ambos nós, sensores e sorvedouros: Escolha do algoritmo de predição e a disseminação do algoritmo de predição escondido. Os experimentos foram conduzidos com o limiar variando de 0,5 a 0,95, com agregação e sem agregação. Na análise dos resultados, constatou-se que ao utilizar o esquema de agregação (sem fazer previsões), o consumo de energia foi reduzido para 60% do total, porém o tamanho dos pacotes enviados é maior. Os maiores ganhos, no entanto, são obtidos através da adoção do esquema de dupla predição somado ao esquema de agregação, economizando até 92% da energia consumida pelas transmissões.

No trabalho de Monteiro et al. (2017), é proposto um esquema de dupla predição para aplicações de monitoramento de variáveis físicas, chamado *Dual Prediction*

³ <http://db.lcs.mit.edu/labdata/labdata.html>

with Cubic Adaptive Sampling (DPCAS). A técnica proposta utiliza um modelo de previsão baseado em séries temporais exponenciais e incorpora um método guloso de amostragem adaptativa baseado em um algoritmo de controle de congestionamento, chamado CUBIC TCP (HA; RHEE; XU, 2008).

A técnica de amostragem adaptativa consiste em variar o intervalo de tempo entre as detecções dos nós sensores de forma a adaptar-se dinamicamente à variabilidade dos fenômenos detectados, visando aumentar o tempo de hibernação dos sensores e reduzir o tráfego de dados na rede. Ou seja, em períodos em que o fenômeno físico apresenta pouca variação, o intervalo entre as detecções pode ser aumentado. Quando o fenômeno físico apresenta maior variação, o intervalo pode ser reduzido.

Ao utilizar a técnica de amostragem adaptativa, os dados coletados pelos sensores não apresentam frequência regular. Por isso, para realizar a previsão de valores futuros, os autores Monteiro et al. (2017) utilizaram uma técnica chamada Extensão de Wright ao método de Holt (WRIGHT, 1986), que é uma adaptação do algoritmo *Double Exponential Smoothing* (DES)⁴, também conhecido como Método Holt, cujo objetivo é utilizar dados históricos irregulares para realizar a previsão valores futuros.

Os experimentos realizados no trabalho de Monteiro et al. (2017) tiveram como objetivo avaliar a acurácia das previsões, bem como o consumo de energia dos nós. Além disso, o desempenho do método proposto foi avaliado através da comparação com os algoritmos *Double Smoothing-based Adaptive Sampling* (EDSAS) (GUPTA et al., 2011) e *Adaptive Sampling - Transmission Control Protocol* (AS-TCP) (AL-HOQANI; YANG, 2015).

As simulações foram realizadas por rotinas desenvolvidas na linguagem de programação C. O consumo de energia dos nós, por sua vez, foi calculado através de um modelo de energia simplificado, proposto por Shnayder et al. (2004). Como entrada para todas as simulações, foram utilizados os dados do Laboratório *Intel Berkeley Research* (MADDEN, 2004). Especificamente, foram utilizados os dados dos sensores de temperatura entre os dias 1 e 7 de Março de 2004.

De acordo com os autores Monteiro et al. (2017), os resultados obtidos foram promissores com relação à qualidade dos dados gerados e à redução do consumo de energia nos nós sensores. Na comparação com as técnicas de previsão *Double Smoothing-based Adaptive Sampling* (EDSAS) e *Adaptive Sampling - Transmission Control Protocol* (AS-TCP), foram obtidos resultados que apresentam um bom equilíbrio entre a qualidade dos dados gerados e o consumo de energia nos sensores.

Já no trabalho de Almhana, Choulakian e Almhana (2017), é proposto um esquema composto por duas abordagens: Predição de dados e análise da dispersão.

⁴ Em Português: Suavização Exponencial Dupla

Para a predição de valores futuros, o modelo ARIMA foi empregado. Este modelo é caracterizado pela junção dos modelos auto-regressivo (AR) e médias móveis (MA), juntamente com um processo chamado de diferenciação, utilizado em séries temporais não estacionárias. A abordagem baseada na análise da dispersão, por outro lado, visa evitar a realização de transmissões de dados caso a dispersão aferida entre duas detecções seja menor que um limiar pré-definido.

O esquema proposto por Almhana, Choulakian e Almhana (2017) leva em conta a arquitetura de uma Rede de Sensores Sem Fio contendo mais de 20.000 nós sensores, voltada para monitoramento do consumo de água, na cidade de Moncton, Canadá. Na abordagem baseada na análise da dispersão, o limiar denominado μ representa uma certa porcentagem do consumo médio, que é a média de n medições de consumo de água precedentes à leitura atual. Este limiar é utilizado a fim de garantir a transmissão de detecções cujos valores demonstrem possíveis vazamentos de água.

Os experimentos realizados pelos autores contaram com cerca de um ano de dados provenientes de mais de 20.000 sensores de medição de consumo de água, sendo aproximadamente 4 detecções por dia. Os resultados obtidos mostraram que o esquema proposto por Almhana, Choulakian e Almhana (2017) atingiu uma redução substancial na transmissão de dados se comparado com o método tradicional de coleta de dados utilizado pela cidade de Moncton, no qual nenhuma estratégia para a redução de transmissões de dados é aplicada.

No trabalho de Fathy, Barnaghi e Tafazolli (2018) foi proposto um método que objetiva reduzir o número de transmissões entre nós sensores e nós sorvedouros para estender o tempo de vida da rede. Essa redução se dá através de um modelo de predição de futuras detecções dos nós sensores. Dessa forma, entende-se que um dado previsto não precisa ser enviado pelo sensor, caso a acurácia do dado previsto seja atestada, não afetando a qualidade do real valor detectado.

O método proposto pelos autores é baseado no algoritmo de filtragem adaptativa *Least Mean Square* (LMS)⁵, combinando dois filtros LMS com diferentes tamanhos para estimar os valores detectados. Essa estimativa é realizada tanto nos nós sensores quanto nos nós sorvedouros, de modo que os nós sensores precisem transmitir apenas seus valores que se desviam significativamente dos valores previstos.

Para determinar se um valor detectado será enviado ou não, utiliza-se um limiar pré-definido, chamado de e_{max} . Deste modo, se o desvio é maior que o limiar ($> e_{max}$) o real valor é enviado pelo sensor. Caso contrário, o envio não é realizado pelo nó sensor e o nó sorvedouro encaminha o valor proveniente da predição. Com a utilização do limiar, esta abordagem pode ser utilizada tanto em aplicações críticas

⁵ Em Português: Mínimo Quadrado Médio

(que requerem maior acurácia nos dados detectados), como não críticas (que não requerem acurácia tão alta), uma vez que o valor do limiar e_{max} pode ser customizado (FATHY; BARNAGHI; TAFAZOLLI, 2018).

Nos experimentos realizados pelos autores Fathy, Barnaghi e Tafazolli (2018), foi utilizado o conjunto de dados do Laboratório *Intel Berkeley Research* (MADDEN, 2004). Durante os experimentos, foram usados especificamente os dados dos sensores de temperatura entre os dias 6 e 9 de Março de 2004, a fim de obter uma comparação justa e consistente com o trabalho de Santini e Romer (2006), que propõe uma abordagem similar, mas com apenas um filtro LMS. Constatou-se que com o limiar $e_{max} = 0,5$, a abordagem proposta alcançou alta redução nas transmissões de dados (em torno de 95%), 5% a mais que no trabalho de Santini e Romer (2006). Embora os resultados da redução das transmissões de dados sejam promissores, não foi realizado um estudo sobre o impacto energético do uso dos algoritmos de predição nos nós sensores e sorvedouros.

3.4 MODELOS BASEADOS EM *COMPRESSIVE SENSING*

No trabalho de Masoum, Meratnia e Havinga (2013), é proposta uma abordagem para o sensoriamento compressivo baseado na dispersão dos dados coletados em Redes de Sensores sem Fio. Esta abordagem utiliza o princípio de amostragem adaptativa, a fim de transmitir apenas as detecções que indicam uma mudança significante no ambiente sensoreado. Essa técnica tem o objetivo de reduzir a sobrecarga na comunicação, e consequentemente, aumentar o tempo de vida da rede. Pressupondo-se que as RSSF geralmente são densas, isto é, possuem grande quantidade de nós, essa abordagem tira proveito da correlação espacial e temporal entre os dados coletados pelos sensores a fim de alcançar maior compressão.

Para definir o conjunto de detecções a ser enviado pelo sensor, é utilizado um modelo estocástico baseado na técnica da Inferência Bayesiana. No cenário de aplicação, os autores Masoum, Meratnia e Havinga (2013) supõe que um sorvedouro deve possuir uma unidade de processamento de alta performance, com fonte de energia considerável. Dessa forma, o sorvedouro é capaz de reconstruir os dados através do Algoritmo de Propagação de Crença (SHAHRASBI; TALARI; RAHNAVARD, 2011).

Os experimentos foram realizados através de simulações no MATLAB⁶. Na simulação, foi definida uma RSSF com 10 nós sensores organizados em uma topologia estrela. Nessa rede foi utilizado apenas um nó sorvedouro localizado na parte central, caracterizando-se um esquema *single-hop* (um salto). Foram realizadas comparações da abordagem proposta com os algoritmos Basis Pursuit (MOTA et al., 2011) e Joint-

⁶ <https://www.mathworks.com/products/matlab.html>

BP (TZAGKARAKIS; STARCK; TSAKALIDES, 2011). Na simulação foi utilizado o *dataset* de detecções de temperatura coletado em uma RSSF localizada na ponte Humber, na cidade de Lincolnshire - Inglaterra⁷. Os resultados demonstram que a abordagem proposta alcançou maior economia de energia e maior acurácia na reconstrução dos dados em comparação com os demais algoritmos.

Huang et al. (2017) propuseram em seu trabalho, um esquema compressivo e adaptativo de coleta de dados. Esse esquema é baseado na detecção adaptativa, que é o ajuste adaptativo da frequência em que as detecções são realizadas pelo sensor. Dessa forma, as detecções podem ser adaptadas de acordo com a variabilidade (dispersão) dos dados coletados pelos sensores. Além disso, os autores propuseram um algoritmo para reconstrução dos dados, com o intuito de maximizar a acurácia nos dados reconstruídos.

Como a detecção adaptativa depende da dispersão dos dados, os autores propuseram também um algoritmo de predição da tendência da dispersão dos dados para que a frequência das detecções também seja definida adaptativamente de acordo com as tendências previstas. Para a reconstrução dos dados, por sua vez, foi utilizado uma versão adaptativa do algoritmo guloso chamado *Stagewise Orthogonal Matching Pursuit algorithm* (StOMP) (TROPP; GILBERT, 2007), que usa o método *Proportional-Integrative-Derivative* (PID) (ANG; CHONG; LI, 2005) para ajustar o limite de dispersão (parâmetro) baseado no erro da reconstrução.

Os experimentos foram executados na plataforma Crossbow Micaz WSN (ALI; DRIEBERG; SEBASTIAN, 2011). Foram realizadas comparações da abordagem proposta com o algoritmo de detecção compressiva canônico e o algoritmo proposto em (QUER et al., 2010). Os resultados mostraram que o esquema de detecção adaptativa alcançou maior acurácia na reconstrução dos dados e maior redução do custo energético dos nós da rede.

No trabalho de Chen et al. (2018a), é proposto um algoritmo, chamado *Dynamic Topology Compression Sensing* (DTCS), que visa realizar a coleta de dados com foco na detecção compressiva em Redes de Sensores sem Fio (RSSF). O algoritmo *Dynamic Topology Compression Sensing* (DTCS) atua em duas etapas: Construção de uma árvore de coleta e coleta compressiva de dados. Na primeira etapa, uma árvore de coleta de dados é construída utilizando um método proveniente da Teoria dos Grafos, chamado Conjunto Independente Máximo. Em seguida, na segunda etapa, os nós sensores realizam comunicações a fim de obter informações sobre a topologia atual da rede. Dessa forma, de acordo com a mudança da topologia, a técnica de *Compressive Sensing* é aplicada, transferindo os dados coletados pelos sensores para o nó sorvedouro.

⁷ <http://www.bridgeforum.com/humber/hessle-anchorage-cal.php>

Os autores Chen et al. (2018a) realizaram simulações com uma plataforma de *testbeds*, chamada BUAA, mantida pela Universidade de Beihang (YI et al., 2013). Inicialmente foram utilizados 200 sensores. Posteriormente, sensores foram incluídos e removidos, a fim de testar a performance da abordagem proposta de acordo com a dinamicidade da rede. Paralelamente, foram realizadas comparações com os trabalhos (LUO; XIANG; ROSENBERG, 2010) e (ZHENG et al., 2013), nas quais foram obtidos resultados promissores quanto à redução e reconstrução dos dados.

3.5 COMPARAÇÃO DOS TRABALHOS RELACIONADOS

Nesta seção encontra-se uma tabela comparativa entre os trabalhos apresentados neste capítulo. Pode-se perceber que na maioria dos trabalhos existe um *trade-off* (conflito de escolha) a ser analisado. Nos trabalhos baseados em predição, existe um *trade-off* entre a acurácia mínima requerida da predição e a quantidade de redução alcançada na transmissão das detecções. Quanto maior for o valor da acurácia mínima requerida, menor será a quantidade de detecções enviadas (maior redução nas transmissões de dados), porém, o erro no conjunto de dados final será maior. O mesmo acontece nos trabalhos baseados em *Compressive Sensing*. Quanto maior o tamanho da amostragem, menor será a redução nas transmissões, porém o conjunto de dados final apresentará um índice menor de erros. Nos trabalhos baseados em Agregação de dados e Fusão de dados, contudo, o *trade-off* é relacionado ao atraso na entrega das mensagens. Quanto maior o atraso, mais mensagens podem ser agregadas/fundidas.

Com relação a confiabilidade dos dados, aplicações críticas que exigem alta confiabilidade, tais como aplicações de cuidado à saúde, aplicações de monitoramento de desastres e aplicações industriais, devem requerer baixos índices de erro no conjunto de dados final, abrindo mão de um alto ganho na redução das detecções. Já aplicações não críticas podem requerer maior redução nas transmissões, bem como, maior erro no conjunto de dados final.

A comparação dos trabalhos relacionados é apresentada na Tabela 1, levando em consideração a classificação do método de redução, o método em si, e o modelo de reconstrução, se existir.

Na aplicação da técnica de *Compressive Sensing*, os algoritmos de reconstrução de dados podem exigir poder computacional e consumo de energia consideráveis nos sorvedouros, o que pode se tornar uma limitação deste modelo. O mesmo ocorre na técnica de redução de dados baseada em predição, uma vez que, quanto maior a quantidade de nós sensores na RSSF, maior será o trabalho do sorvedouro de realizar previsões futuras para cada um dos nós, baseado em seus dados históricos.

Tabela 3.1 – Comparação dos Trabalhos Relacionados

Trabalhos Relacionados	Classificação	Método	Reconstrução
(DIAS; BELLALTA; OECHSNER, 2016b)	Híbrida	Predição / Agregação	-
(MONTEIRO et al., 2017)	Local	Predição	-
(ALMHANA; CHOULAKIAN; ALMHANA, 2017)	Local	Predição	-
(FATHY; BARNAGHI; TAFAZOLLI, 2018)	Local	Predição	-
(SANTINI; ROMER, 2006)	Local	Predição	-
(XU et al., 2015)	Híbrida	Agregação / Compressão	DCT
(ZEGARRA et al., 2016)	Global	Agregação	-
(LI et al., 2017)	Global	Agregação	-
(CALLEGARO, 2014)	Híbrida	Fusão / Agregação	-
(ZOU; LIU, 2015)	Local	Fusão	-
(MASOUM; MERATNIA; HAVINGA, 2013)	Local	Compressão	APC
(HUANG et al., 2017)	Local	Compressão	StOMP
(CHEN et al., 2018a)	Híbrida	Compressão / Agregação	DTCS
Abordagem Proposta	Local	Análise da dispersão	Tendência Assintótica

Os trabalhos baseados em agregação de dados, por sua vez, são diferenciados pelo fato de efetuar a redução de dados após o seu envio. Dessa forma, a redução de dados pode ser atestada na diminuição da sobrecarga da rede ao agregar dados de diferentes sensores até chegar ao sorvedouro, ao invés de simplesmente retransmiti-los. Uma vez que um nó sensor precisa agregar dados de vários outros sensores, a latência na entrega das mensagens pode se tornar uma limitação deste modelo para aplicações críticas em relação ao atraso.

Já os trabalhos baseados em Fusão de dados, efetuam a redução de dados tanto no sensor quanto na rede. A Fusão de dados de dados explora a correlação existente entre os dados obtidos por vários sensores em um único nó, além de detectar a redundância de dados na fusão em nós vizinhos.

O método utilizado para a redução de dados proposto neste trabalho é baseado na análise da dispersão dos dados coletados pelos sensores de forma a realizar

transmissão de dados apenas quando houver mudança significativa no ambiente sensoriado. Dessa forma é possível reduzir consideravelmente a quantidade de transmissões na rede, sem exigir um alto grau de computação dos nós.

A maioria dos trabalhos relacionados apresenta a aplicação de experimentos em simuladores de rede, ou plataformas de *testbeds*, a fim de avaliar o consumo de energia dos nós, bem como a sobrecarga na comunicação da rede. Dentre os simuladores utilizados pelos trabalhos, têm-se: Omnet++, Sinlago, SIDnet-SWANS e MATLAB. Dentre as plataformas de *testbeds*, têm-se: Crossbow Micaz e TelosB. Neste trabalho, foram realizados experimentos com o simulador Castalia (RASTEGARNIA; SOLOUK, 2011), que é um simulador baseado na plataforma Omnet++ e foi concebido especificamente para simulações de RSSF, redes de área corporal⁸ e redes de dispositivos de baixa potência.

3.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram apresentados trabalhos que lidam com o problema da redução da transmissão de dados em nós sensores, encontrados na literatura. Dentre as técnicas encontradas, destacam-se: Redução de dados baseada em predição, Agregação de Dados, Fusão de dados e *Compressive Sensing*. A abordagem proposta neste trabalho de dissertação, contudo, é baseada na análise da dispersão dos dados coletados pelos sensores, identificando e deixando de enviar valores que demonstrem baixa dispersão.

Uma vez que a abordagem proposta realiza a redução de dados no sensor, antes dos dados serem enviados, ela se assemelha às técnicas de redução de dados baseada em predição e *Compressive Sensing*. Contudo, tais técnicas podem exigir alto poder de processamento e consumo de energia do nó sorvedouro. A abordagem proposta, contudo, visa superar estas limitações, sem exigir alto grau de computações no sorvedouro.

No próximo capítulo é apresentada a abordagem para redução das transmissões de dados em nós sensores proposta neste estudo. A abordagem é baseada na análise da dispersão (variabilidade) dos valores detectados pelos sensores. O objetivo principal é evitar que valores muito semelhantes sejam transmitidos. Essa abordagem oferece algoritmos para a redução de dados baseada na análise da dispersão, bem como para a reconstrução de dados no nó sorvedouro.

⁸ Do Inglês: *Body Area Networks* (BAN)

4 ABORDAGEM PARA REDUÇÃO DAS TRANSMISSÕES DE DADOS EM RSSF

Este capítulo descreve o funcionamento do método proposto para redução das transmissões de dados em sensores operando em aplicações de monitoramento. O objetivo principal da proposta é evitar que nós sensores transmitam detecções cujos valores demonstram baixa dispersão (variabilidade/espalhamento) dentro de um limite aceitável. A proposta também conta com um método de reconstrução baseado na tendência assintótica das detecções que deixaram de ser enviadas pelos nós sensores. Como resultado, espera-se uma série temporal reconstruída com o mínimo de prejuízos em relação a série temporal original, a partir de um número menor de detecções transmitidas para a rede.

Nas aplicações que utilizam sensores para realizar monitoramento, como as Redes de Sensores Sem Fio (RSSF), os nós sensores geralmente efetuam detecções de variáveis do ambiente físico com uma frequência pré-definida. O resultado disto é a criação de uma coleção ordenada de dados. Para exemplificar, essa coleção pode ser chamada de S , iniciando em um ponto no tempo t_0 , e seguindo até um instante de tempo final t_n . Especificamente, $S = \{s_0, s_1, s_2, \dots, s_n\}$, na qual o n -ésimo elemento é um par $s_n = (V_n, t_n)$. O primeiro elemento do par, V_n , indica o valor da variável detectada pelo sensor (por exemplo: valor de temperatura), enquanto que o segundo elemento do par, t_n , indica o instante de tempo da ocorrência da leitura.

4.1 ANÁLISE DA DISPERSÃO DE VALORES DETECTADOS EM NÓS SENSORES

De acordo com (EL-TELBANY; MAGED, 2017), as detecções realizadas pelos nós sensores em aplicações de monitoramento frequentemente retornam valores muito semelhantes, ou até mesmo idênticos. O método proposto neste trabalho é baseado na análise da dispersão dos dados coletados pelos nós sensores, que visa identificar conjuntos de detecções com valores que demonstrem baixa dispersão, ou seja, valores próximos uns dos outros.

A abordagem para redução na transmissão de dados em nós sensores funciona como exposto a seguir. Suponha que um conjunto de detecções (duas ou mais) de um determinado nó sensor demonstrem um certo nível de similaridade (baixa dispersão). O nó sensor armazena esses valores em caráter temporário, visando não transmitir todas as detecções com valores pouco dispersos. A seguir, o nó sensor transmite uma mensagem ao sorvedouro contendo apenas a primeira e última detecções deste conjunto, juntamente com a quantidade de detecções não enviadas. Estes dados serão utilizados em um processo de reconstrução a ser realizado no nó sorve-

douro, no qual a primeira e última detecções deste conjunto são utilizados como base para construção de uma nova série temporal. A partir de então, os valores que estavam guardados no nó sensor podem ser removidos, e a análise dos valores detectados é retomada a fim de encontrar novos conjuntos de detecções com baixa dispersão.

A medida de dispersão é representada por um número real não-negativo que é zero quando os dados são iguais. Esse número aumenta à medida em que os dados se divergem (NATRELLA, 2010). No contexto da abordagem proposta neste trabalho, a análise da dispersão é realizada a cada detecção dos nós sensores, e a medida de dispersão é a diferença obtida na subtração entre os valores absolutos de V_i e V_{i-1} .

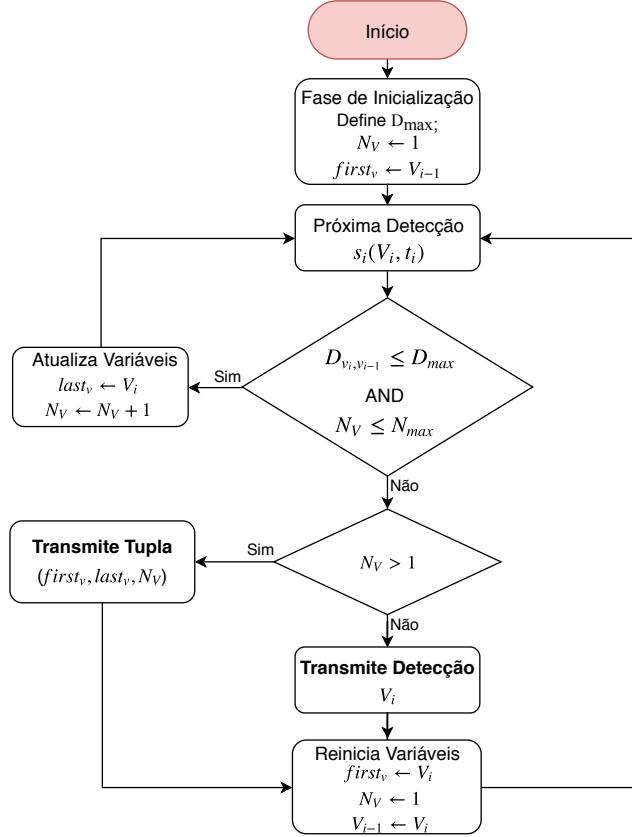
As operações de subtração entre valores absolutos correspondem à uma norma chamada l_1 , também chamada de Distância de *Manhattan*, proveniente da Geometria do Táxi (KRAUSE, 1986), uma vez que mede a distância entre dois pontos de forma análoga à distância percorrida em uma cidade ao longo de quarteirões ortogonais. Justifica-se o uso da norma l_1 (Distância de *Manhattan*) como medida de dispersão pela simplicidade na implementação e pelo baixo custo computacional. A Distância de Manhattan é calculada através da Equação 4.1.

$$D_{V_i, V_{i-1}} = |V_i - V_{i-1}| \quad (4.1)$$

Com o objetivo de permitir que o nó sensor seja capaz de identificar um conjunto de detecções com baixa dispersão, define-se um parâmetro, denominado D_{max} , que representa o limite máximo aceitável para o distanciamento entre os valores detectados. Mediante o exposto, se a distância entre o valor das detecções V_i e V_{i-1} for menor ou igual a D_{max} , então a detecção V_i não será transmitida, mas será armazenada temporariamente em uma variável, denominada $last_V$. Além disso, um contador N_V que inicia em 1 é incrementado. Este procedimento será repetido enquanto a condição $D_{V_i, V_{i-1}} \leq D_{max}$ for satisfeita. Quando tal condição não for satisfeita, uma tupla é transmitida, contendo a primeira e última detecções do conjunto, armazenadas nas variáveis $first_v$ e $last_v$, respectivamente. A variável N_V também será enviada como um terceiro elemento da tupla, uma vez que será utilizada no processo de reconstrução de dados, discutido na Seção 4.2.

Em determinadas situações, é possível que o número de detecções com valores pouco dispersos em um nó sensor seja muito grande. Para evitar que o nó sensor armazene uma quantidade muito grande de detecções, bem como evitar um alto atraso no envio de mensagens, define-se um parâmetro que estabelece a quantidade máxima de detecções em um conjunto de dados com baixa dispersão, denominado N_{max} . O fluxo completo deste esquema é exibido na Figura 4.1.

Figura 4.1 – Fluxo de execução do Algoritmo de Reconstrução



A fase de inicialização, expressa na Figura 4.1, comprehende a coleta das primeiras detecções do nó sensor. Esses dados são utilizados para encontrar, de forma automática, o valor ideal para D_{max} . Detalhes sobre a fase de inicialização serão abordados na Seção 4.3.

A escolha do parâmetro N_{max} , por sua vez, depende dos requisitos de cada aplicação de monitoramento com relação ao atraso na entrega das detecções. Quanto menor o atraso desejado, menor deve ser o valor de N_{max} . Consequentemente, menor redução é alcançada. O algoritmo que implementa a abordagem proposta é apresentado no Algoritmo 4.1.

O esquema de funcionamento do algoritmo é exemplificado através da Figura 4.2. Na Figura 4.2(a) são apresentados 19 valores de temperatura coletados por um nó sensor. Neste exemplo, o método proposto não é aplicado. Logo, pode-se verificar que a quantidade de detecções é igual ao número de transmissões realizadas do sensor para a rede. Ou seja, foram realizadas 19 detecções e 19 transmissões.

Ao executar o algoritmo sobre estes dados, com o parâmetro $D_{max} = 0,4$, como ilustrado na Figura 4.2(b), o número de transmissões diminui de 19 para 10. Quando $D_{max} = 0,7$, como na Figura 4.2(c), o número de transmissões diminui de

Algorithm 4.1 Redução de Dados Baseado na Análise da Dispersão

```

1: FirstReadings[]                                ▷ Primeiras detecções do sensor
2:  $N \leftarrow 50$                                 ▷ Número de detecções na fase de inicialização. Ex: 50
3:  $N_V \leftarrow 1$                                 ▷ Contador de detecções não transmitidas
4:  $N_{max} \leftarrow 45$                             ▷ Número máximo de transmissões evitadas. Ex: 45
5:  $D_{max} \leftarrow \text{GETDMAXVALUE}(\text{firstReadings}[], N)$ 
6:  $V_{i-1} \leftarrow \text{FirstReadings}[N - 1]$ 
7:  $firstV \leftarrow V_{i-1}$                             ▷ Define valor de  $D_{max}$ 
8: while True do
9:    $V_i \leftarrow \text{GETSENSORVALUE}()$ 
10:  if  $|V_i - V_{i-1}| \leq D_{max}$  and  $N_V < N_{max}$  then
11:     $lastV \leftarrow V_i$ 
12:     $N_V \leftarrow N_V + 1$ 
13:  else
14:    if  $N_V > 1$  then
15:       $\text{SENDDATATOSINK}((firstV, lastV, N_V))$           ▷ Transmissão de dados (Tupla)
16:       $firstV \leftarrow V_i$ 
17:    else
18:       $\text{SENDDATATOSINK}((V_{i-1}))$                   ▷ Transmissão de dados (Detecção isolada)
19:       $firstV \leftarrow V_i$ 
20:    end if
21:     $N_V \leftarrow 1$ 
22:  end if
23:   $V_{i-1} \leftarrow V_i$ 
24: end while
  
```

Figura 4.2 – (a) Série original; (b) Série manipulada, onde $D_{max} = 0, 4$; (c) Série manipulada, onde $D_{max} = 0, 7$. Fonte: Próprio Autor

(a)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	11,8	12,1	12,4	12,3	12,4	11,8	7,12	12,6	12,7	12,8	12,4	12,3	13,2	12,8	2,12	12,6	13,4	12,7	12,8
(b)	1	2	3	4	5	6	7	8	9	10									
	11,8; 12,4; 5		11,8	7,12	12,6; 12,8; 3	12,4; 12,3; 2	13,2; 12,8; 2	2,12	12,6	13,4	12,7; 12,8; 2								
(c)	1	2	3	4	5	6	7	8											
	11,8; 12,4; 6		7,12	12,6; 12,3; 5	13,2; 12,8; 2	2,12	12,6	13,4	12,7; 12,8; 2										

19 para 8. Nesse caso, a redução da quantidade de transmissões é maior que de 50%. Quanto maior o valor de D_{max} , maior a redução nas transmissões de dados. Do mesmo modo, quanto menor o valor de D_{max} , menor será a redução alcançada.

4.2 ALGORITMO PARA RECONSTRUÇÃO DOS DADOS

Uma questão importante no processo de redução de dados, é garantir que, no seu destino, os dados enviados pelos nós sensores apresentem a frequência correta em que foram detectados. Baseado neste fato, é proposto um Algoritmo de Reconstrução de Dados, apresentado no Algoritmo 4.2.

Algorithm 4.2 Reconstrução dos Dados

```

1: function EVENLYSPACEDVALUES(start, end, num)
2:   output = []
3:   delta  $\leftarrow$  (end - start)/(num - 1)
4:   for (i  $\leftarrow$  0 ; i < num ; i  $\leftarrow$  i + 1) do
5:     output.append(start + delta * i)
6:   end for
7:   output.append(end)
8:   return output
9: end function
10: while True do
11:    $V_i \leftarrow$  RECEIVESENSORDATA()
12:   if  $V_i$  is Tuple then
13:      $temp \leftarrow$  EVENLYSPACEDVALUES( $V_i[0], V_i[1], V_i[2]$ )
14:     for i <  $V_i[2]$  do
15:       dataset.append(temp[i])
16:     end for
17:   else
18:     dataset.append(V_i)
19:   end if
20: end while
  
```

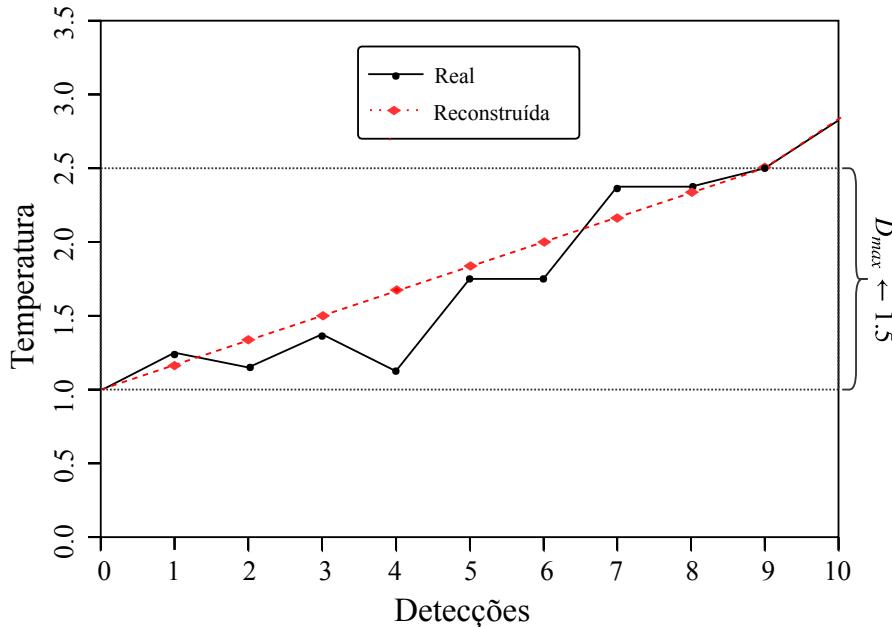
O Algoritmo 4.2 realiza a reconstrução dos dados baseando-se na tendência assintótica de um determinado conjunto de detecções com baixa dispersão. Em outras palavras, o Algoritmo para Reconstrução dos Dados é responsável por definir valores uniformemente espaçados entre dois limites a fim de substituir os valores que não foram transmitidos pelo nó sensor. Os limites em questão são representados pelo primeiro e último elemento da tupla enviada pelo nó sensor ($first_v$ e $last_v$).

Assumindo que o nó sorvedouro receba os dados otimizados pelo nó sensor, como descrito no Algoritmo 4.1, verifica-se se a mensagem recebida contém uma detecção isolada ou uma tupla (linha 12). Quando a mensagem recebida se trata de uma tupla, valores uniformemente espaçados serão criados entre $V_i[0]$ e $V_i[1]$, de acordo com o terceiro elemento da tupla $V_i[2]$. Dessa forma os dados reconstruídos consistirão na mesma quantidade das detecções realizadas pelo nó sensor, embora nem todas tenham sido transmitidas. Na Figura 4.3 é ilustrado um exemplo de reconstrução de dados baseado na tendência assintótica.

Conforme ilustrado na Figura 4.3, a linha escura com marcações em círculos representa os dados reais. A linha vermelha com marcações em losangos representa os dados reconstruídos. Nesse exemplo, o valor de D_{max} foi definido em 1,5. Dessa forma, os valores reais detectados desde o instante de tempo 0 até o instante de tempo 9 (eixo x) formam um conjunto de detecções pouco dispersas, já que a dispersão entre elas é menor que o valor de D_{max} .

Considerando os dados apresentados na Figura 4.3, a tupla recebida pelo algoritmo de reconstrução contém os seguintes valores: (1,0; 2,5; 10), sendo que o

Figura 4.3 – Tendência Assintótica. Fonte: Próprio Autor



primeiro elemento é representado pela primeira detecção deste conjunto de detecções cuja dispersão é menor que o valor de D_{max} . O segundo elemento é representado pela última detecção deste conjunto. O último elemento, por sua vez, é representado pela quantidade total de detecções deste conjunto, incluindo a primeira (instante de tempo 0) e última (instante de tempo 9) detecções.

Isto significa que o algoritmo de reconstrução irá calcular valores uniformemente espaçados entre 1,0 e 2,5 a fim de reconstruir a série temporal, respeitando a tendência da série temporal original.

Embora a reconstrução baseada na tendência assintótica gere valores de forma uniforme, representando um determinado nível de perdas com relação à série temporal original, identificou-se que o erro obtido com este modelo é menor do que com outros, como por exemplo, a substituição das detecções não enviadas por médias.

4.3 FASE DE INICIALIZAÇÃO PARA A ESCOLHA DO VALOR DE D_{max}

Para que a escolha de D_{max} seja realizada de forma adequada, é necessário que haja um conhecimento a-priori sobre os dados coletados em um nó sensor. Através desse conhecimento, é possível perceber a medida em que os dados coletados se dispersam. Por exemplo, considerando os exemplos mostrados na Figura 4.2(a),

a diferença entre os valores menos dispersos varia entre 0,1 e 1. Portanto, qualquer valor entre 0,1 e 1 pode ser escolhido para D_{max} , sendo que quanto maior o valor, maior redução pode ser alcançada, como mostrado nas Figuras 4.2(b) e 4.2(c).

Para que essa escolha aconteça de forma automática, é necessário a definição de um parâmetro que represente o erro máximo aceitável por uma determinada aplicação de monitoramento. Através deste parâmetro, denominado e_{max} , o valor de D_{max} poderá ser escolhido, a fim de que os dados finais não apresentem erro superior ao máximo permitido.

Considerando o exposto, um algoritmo que analisa as primeiras detecções do nó sensor é proposto. Neste estudo foram consideradas as primeiras 50 detecções do nó sensor. As estratégias para esta análise são apresentadas no Algoritmo 4.3.

Com base nas primeiras detecções, o algoritmo gera um conjunto de dados reduzido (linha 69) através de uma adaptação do Algoritmo 1, com D_{max} iniciando com um valor consideravelmente baixo, a fim de que seja elevado até que se encontre um valor ideal. Em seguida, esse novo conjunto de dados é reconstruído (adaptação do Algoritmo 2) localmente no sensor, e então compara-se o conjunto de dados reconstruído com o conjunto de dados inicial. Caso o erro obtido na reconstrução dos dados seja menor que o erro máximo aceitável (e_{max}), repete-se o procedimento elevando o valor de D_{max} . Ou seja, o valor de D_{max} será elevado até que o erro obtido na reconstrução seja igual ou superior ao parâmetro e_{max} .

É importante enfatizar que, ao elevar o valor de D_{max} , é possível que o erro obtido na reconstrução não seja alterado. Tal fato pode decorrer em consequência do uso do parâmetro N_{max} , apresentado na Seção 4.1, que limita o tamanho de conjuntos de detecções com baixa dispersão, reduzindo o potencial de redução a ser alcançado. Dessa forma, caso o erro obtido não seja elevado ao ponto de equivaler ao erro máximo aceitável (e_{max}), a estrutura de repetição (*while*) pode ser finalizada após uma determinada quantidade de iterações, por exemplo, 100 (linha 66). Esse controle é realizado através das variáveis *flag* e *counter*, impedindo o algoritmo de permanecer em *loop* indefinidamente.

O erro obtido na reconstrução dos dados é determinado através do cálculo do Erro Absoluto Médio (EAM) (WILLMOTT; MATSUURA, 2005). O cálculo do Erro Absoluto Médio funciona da seguinte forma: suponha que um determinado conjunto de dados real é representado por a , e o conjunto de dados reconstruído é representado por b . O Erro Absoluto Médio de b é calculado através da Equação 4.2.

$$EAM = \frac{\sum_{i=0}^n |a_i - b_i|}{n} \quad (4.2)$$

Quando o valor do EAM obtido for igual ou maior que o erro máximo aceitável (e_{max}), o valor final de D_{max} será retornado (linha 80).

Pode se perceber que a definição de um valor para o erro máximo aceitável (e_{max}) é crucial para que o valor de D_{max} seja escolhido na fase de inicialização. (KARIM, 2017) destaca que algumas aplicações de monitoramento são críticas, como por exemplo, detecção de fogo ou vazamento de gás. Essas aplicações exigem alta acurácia nos dados, bem como um baixo nível de atraso na entrega das mensagens. Logo, o valor escolhido para o erro máximo aceitável nesses casos deve ser baixo (próximo de zero). Ou seja, pouca redução será alcançada, dados os requisitos da aplicação.

Na Tabela 4.1 são listadas as variáveis utilizadas nos Algoritmos 4.1, 4.2 e 4.3.

Tabela 4.1 – Parâmetros e Variáveis utilizados nos Algoritmos 4.1, 4.2 e 4.3

D_{max}	Distância máxima para que duas detecções sejam consideradas pouco dispersas
N_{max}	Quantidade máxima de detecções em um conjunto de detecções de baixa dispersão
e_{max}	Erro máximo aceitável (Varia de acordo com os requisitos de cada aplicação)
<i>initial_data</i>	Vetor com as primeiras detecções
N	Quantidade de detecções em vetor
N_V	Contador de detecções em um conjunto de detecções de baixa dispersão
V_i	Valor da detecção (s_i) do nó sensor no instante t_i
<i>reduced_data</i>	Vetor com número reduzido de detecções
<i>reconstructed_data</i>	Vetor com dados reconstruídos
<i>EAM</i>	Erro absoluto médio

Pode-se perceber no Algoritmo 4.3 que o valor de D_{max} a ser retornado é decrementado em 0,02. Este decremento se dá pelo fato de que o valor de D_{max} já havia sido incrementado (linha 80) antes que a condição $MAE \leq e_{max}$ não fosse satisfeita. Por isso, quando tal condição não for satisfeita, é necessário decrementar o valor de D_{max} a fim de que o erro nos dados finais não excedam o erro máximo aceitável.

Com relação ao ajuste dos parâmetros e_{max} e N_{max} , foi criada uma taxonomia baseada no trabalho de Yetgin et al. (2017) que classifica diferentes tipos de aplicação de acordo com seus requisitos, atribuindo valores específicos para e_{max} e N_{max} . Especificamente, foram definidas três classes de aplicações de monitoramento (0, 1 e 2) que são baseadas em seus requisitos de atraso na entrega das mensagens e no erro máximo aceitável na reconstrução dos dados.

A classe 0 é menos tolerante a atrasos e exige maior acurácia nos dados reconstruídos. Nesta classe podem ser incluídas aplicações de monitoramento críticas,

Algorithm 4.3 Fase de Inicialização - Escolha de D_{max}

```

1: function DATAREDUCTION(initial_data[], N,  $D_{max}$ ,  $N_{max}$ )
2:   ReducedData[]
3:    $x \leftarrow 1$                                      ▷ Variável de controle do loop
4:    $N_V \leftarrow 1$ 
5:    $first_v \leftarrow initial\_data[0]$ 
6:   while  $x < N$  do
7:      $V_i \leftarrow initial\_data[x]$ 
8:      $V_{i-1} \leftarrow initial\_data[x - 1]$ 
9:     if  $|V_i - V_{i-1}| \leq D_{max}$  and  $N_V < N_{max}$  then
10:       $lastV \leftarrow V_i$ 
11:       $N_V \leftarrow N_V + 1$ 
12:      if  $x == (N - 1)$  then
13:        ReducedData.append(( $first_v, last_v, N_V$ ))
14:      end if
15:    else
16:      if  $N_V > 1$  then
17:        ReducedData.append(( $first_v, last_v, N_V$ ))
18:        if  $x == (N - 1)$  then
19:          ReducedData.append( $V_i$ )
20:        end if
21:         $firstV \leftarrow V_i$ 
22:      else
23:        ReducedData.append( $V_{i-1}$ )
24:         $firstV \leftarrow V_i$ 
25:        if  $x == (N - 1)$  then
26:          ReducedData.append( $V_i$ )
27:        end if
28:      end if
29:       $N_V \leftarrow 1$ 
30:    end if
31:     $V_{i-1} \leftarrow V_i$ 
32:     $x \leftarrow x + 1$ 
33:  end while
34:  return ReducedData                                ▷ Retorna dados reduzidos
35: end function
36: function EVENLYSPACEDVALUES(start, end, num)
37:   output = []
38:    $delta \leftarrow (end - start)/(num - 1)$ 
39:   for ( $i \leftarrow 0$  ;  $i < num$  ;  $i \leftarrow i + 1$ ) do
40:     output.append(start +  $delta * i$ )           ▷ Criação dos valores uniformemente espaçados
        (tendência)
41:   end for
42:   output.append(end)
43:   return output                                ▷ Retorna valores uniformemente espaçados
44: end function
45: function DATARECONSTRUCTION(ReducedData[], N)
46:   ReconstructedData[]
47:   for  $i < N$  do
48:     if ReducedData[i] is Tuple then
49:       temp  $\leftarrow$  EVENLYSPACEDVALUES(ReducedData[i][0], ReducedData[i][1], ReducedData[i][2])
50:       for ( $i \leftarrow 0$  ;  $i < ReducedData[i][2]$  ;  $i \leftarrow i + 1$ ) do
51:         ReconstructedData.append(temp[i])
52:       end for
53:     else
54:       ReconstructedData.append(ReducedData[i])
55:     end if
56:   end for
57:   return ReconstructedData                      ▷ Retorna dados reconstruídos
58: end function

```

```

59: function GETDMAXVALUE(FirstReadings[], N)
60:    $D_{max} \leftarrow 0.01$                                  $\triangleright$  Valor inicial de  $D_{max}$ . Ex: 0,01
61:    $MAE \leftarrow 0$                                       $\triangleright$  Erro Absoluto Médio (EAM)
62:    $e_{max} \leftarrow 0.5$                                  $\triangleright$  Erro máximo aceitável. Ex: 0,5
63:    $flag \leftarrow 0$ 
64:    $counter \leftarrow 0$ 
65:    $exceeded\_value \leftarrow 0.02$ 
66:   while  $MAE \leq e_{max}$  do
67:     if  $counter > 100$  then                                 $\triangleright$  Controle que impede loops infinitos
68:       break
69:     end if
70:      $ReducedData \leftarrow \text{DATAREDUCTION}(FirstReadings[], N, D_{max})$ 
71:      $ReconstructedData \leftarrow \text{DATARECONSTRUCTION}(ReducedData[], N)$ 
72:      $MAE = (\sum_{i=0}^n |FirstReadings[i] - ReconstructedData[i]|)/n$ 
73:     if  $flag == EAM$  then
74:        $counter \leftarrow counter + 1$ 
75:     else
76:        $counter \leftarrow 1$ 
77:     end if
78:      $D_{max} \leftarrow D_{max} + 0.01$ 
79:      $flag \leftarrow MAE$ 
80:   end while
81:   return  $D_{max} - exceeded\_value$                                  $\triangleright$  Decrementa valor final de  $D_{max}$ 
82: end function

```

como aquelas voltadas à detecção de vazamento de gases, monitoramento da condição de equipamentos (manutenção preditiva), e outras. Nesses casos, a frequência das transmissões de dados costuma ser mais alta, a fim de detectar de maneira célere caso algum evento importante ocorra. A classe 2, por sua vez, exige menor nível de acurácia e é mais tolerante a atrasos. Nesta classe encontram-se as aplicações de monitoramento não-críticas, como por exemplo, monitoramento ambiental (temperatura, umidade e luminosidade). A classe 1, por sua vez, possui valores intermediários entre as classes 0 e 2 e incluem as aplicações de monitoramento que demandam um nível moderado de acurácia nos dados, bem como na tolerância ao atraso, tais como, monitoramento da qualidade da água e monitoramento da umidade do solo, amplamente utilizado na agricultura.

Para exemplificar, considere um conjunto de dados proveniente de uma aplicação de monitoramento de compressores de ar, a fim de realizar a manutenção preditiva desse equipamento. Este é o contexto da aplicação de monitoramento no trabalho de Castañeda (2016). Neste caso, a classe desta aplicação será 0, pois se trata da manutenção preditiva de equipamentos. Nesse contexto, qualquer modificação no ambiente sensoreado pode representar uma possível falha do equipamento e deve ser identificado de maneira célere. Por isso, o valor de N_{max} definido nesse caso é 15, e o valor de e_{max} é definido em 0,25. No próximo capítulo serão apresentados os valores específicos definidos para e_{max} e N_{max} de acordo com cada aplicação de monitoramento cujos conjuntos de dados foram utilizados neste trabalho.

Para avaliar a acurácia do modelo proposto, além de realizar comparações entre conjuntos de dados originais e reconstruídos, foram implementados três modelos de predição de séries temporais. São eles: Modelo ARIMA, SVM e Rede Neural Recorrente.

A análise preditiva pode ser fundamental em aplicações de monitoramento, visando a previsão de valores futuros (SHI-NASH; HARDOON, 2016). Dessa forma, os modelos de predição implementados serão aplicados a conjuntos de dados reais e conjuntos de dados reconstruídos pela abordagem proposta, visando verificar se a redução das transmissões afeta o resultado das predições, bem como, atestar qual dos modelos de predição se ajusta melhor a cada conjunto de dados.

4.4 MODELOS DE PREDIÇÃO DE SÉRIES TEMPORAIS

4.4.1 Modelo *Auto-Regressive Integrated Moving Average (ARIMA)*

Como afirma Marcellino, Stock e Watson (2006), os parâmetros do modelo ARIMA são p , d e q . O primeiro parâmetro, p , é o número de observações de atraso incluídas no modelo, também chamado de ordem de atraso. O segundo parâmetro, d , representa o número de vezes que as observações são diferenciadas, também chamadas de grau de diferenciação. O último parâmetro, q é o tamanho da janela da média móvel, também chamada de ordem da média móvel.

Para realizar as previsões de valores futuros, o algoritmo segue as etapas expressas abaixo:

1. Chamar a função *ARIMA()* e passar os parâmetros p , d e q ;
2. Definir os dados de treinamento através da função *fit()*;
3. Realizar as predições através da chamada da função *forecast()* que realiza a previsão de um passo a frente.

Nos experimentos, apresentados no Capítulo 5, cada *dataset* utilizado foi dividido para os conjuntos de treino e de teste a fim de ajustar o modelo e realizar previsões mais assertivas. 75% dos dados foram definidos para os conjuntos de treino e 25% para os conjuntos de teste.

4.4.2 Modelo *Support Vector Regression (SVR)*

Em problemas de regressão, o método *Support Vector Regression (SVR)* (DRUCKER et al., 1997) pode ser aplicado objetivando a predição de séries temporais. Para realizar previsões através do SVR, é necessário definir a função *kernel* a ser utilizada.

Nesta implementação, utilizou-se o *kernel* de função de base radial, também chamado de *kernel* RBF (RUBIO et al., 2007).

Os parâmetros do modelo SVR são: C , que representa o erro; ϵ , que especifica uma margem de tolerância onde nenhuma penalidade é atribuída aos erros; e por fim, γ , que é o inverso do desvio padrão do *kernel* RBF (função gaussiana), que é usado como medida de similaridade entre dois pontos.

Para realizar as previsões de valores futuros, o algoritmo segue as etapas expressas abaixo:

1. Chamar a função `sklearn.svm.SVR()` e passar os parâmetros C , ϵ e γ ;
2. Ajustar o modelo SVR de acordo com os dados de treinamento fornecidos através da função `fit()`;
3. Realizar as previsões através da chamada da função `predict()` que realiza a previsão de um passo a frente.

4.4.3 Modelo baseado em Rede Neural Recorrente

Em Redes Neurais Recorrentes, existe um custo¹ envolvido no processo de aprendizado. Segundo Shanthamallu et al. (2017), esse custo é representado pelo tamanho do erro obtido de acordo com os valores dos parâmetros que foram utilizados. Efetivamente, custo é o tamanho do desvio obtido, dado um determinado conjunto de parâmetros. Neste caso, é necessário minimizar a função de custo para que a performance do aprendizado seja melhorada.

O método do Gradiente Descendente é utilizado em otimização com a finalidade de minimizar a função custo utilizando um esquema iterativo. Tendo em vista que a função em questão é convexa, a cada iteração se toma a direção negativa do gradiente até que o mínimo global seja alcançado. Porém, este método pode não ser ideal quando se tem um conjunto de dados muito grande, porque neste caso, o algoritmo precisaria percorrer o conjunto de dados por inteiro para computar o gradiente e dar apenas um passo na otimização.

Tendo em vista que os dados de Internet das Coisas são massivos, neste estudo foi utilizado o método de regressão linear por Gradiente Descendente Estocástico. Neste método são criados mini-lotes de dados para serem percorridos pelo algoritmo de forma aleatória. Consequentemente, não é necessário percorrer todos os dados para computar o gradiente e é possível obter uma aproximação apenas percorrendo alguns exemplos do conjunto de dados. É fato que para chegar mais próximo

¹ Do Inglês: Loss

do mínimo, será necessária uma grande quantidade de iterações, porém, o tempo da iteração será consideravelmente curto, e o aprendizado mais rápido (BOTTOU, 2010).

Uma vez que o método Gradiente Descendente Estocástico é utilizado no modelo proposto, é necessário embaralhar os mini-lotes de treino para que a função custo seja otimizada. Dessa forma é possível escolher de forma arbitrária o número de iterações para o conjunto de treino.

Nos testes realizados no Capítulo 5, identificou-se que a quantidade de iterações necessárias para a otimização da função custo varia de acordo com cada conjunto de dados. Por exemplo, alguns conjuntos de dados podem demonstrar comportamentos que exijam alta quantidade de iterações no conjunto de treino para que a otimização da função custo chegue ao mais próximo possível do ótimo. Dessa forma, o parâmetro relacionado ao número de iterações do conjunto de treino foi implementado de forma *online* e determinística, como explica André e Parpinelli (2014). O algoritmo monitora o erro obtido em cada iteração e finaliza o conjunto de treino caso esse valor não diminua em uma determinada quantidade de iterações. Contudo, caso o número de iterações chegue a 10.000, o conjunto de teste é finalizado a fim de evitar o sobreajuste². De acordo com Silver (2013) o sobreajuste acontece quando o modelo de previsão se ajusta muito bem ao conjunto de treino, mas se mostra ineficaz para prever novos resultados.

4.4.4 Métricas para Avaliação dos Valores Previstos

Identificou-se necessária a utilização de métricas para definir o nível de assertividade dos modelos de predição mencionados. Dentre as métricas mais utilizadas em aprendizado de máquina, destaca-se o Erro Absoluto Médio. O Erro Absoluto Médio mede o afastamento médio das previsões em relação aos valores observados, focando a estimativa na mediana condicional. Willmott e Matsuura (2005) realizaram um estudo demonstrando vantagens na utilização desta métrica sobre outras, como erro quadrático médio.

Outra métrica utilizada nos modelos ARIMA, SVM e RNN é o coeficiente de determinação R^2 . O coeficiente de determinação, é uma medida de ajustamento de um modelo estatístico linear generalizado, como a regressão linear, em relação aos valores observados. O valor de R^2 varia entre 0 e 1, indicando, em porcentagem, o quanto o modelo consegue explicar os valores observados (NAGELKERKE, 1991).

² Do Inglês: *Overfitting*

4.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi apresentado o algoritmo para a redução da transmissão de dados em Redes de Sensores sem Fio, baseado na análise da dispersão dos dados coletados pelos nós sensores. O objetivo principal deste algoritmo é evitar a transmissão de detecções que demonstrem um determinado nível de similaridade (baixa dispersão). Além disso, foi apresentado o algoritmo para a reconstrução dos dados no destino final, que tem a função de adicionar à série temporal uma aproximação dos dados que deixaram de ser enviados pelos nós sensores. Tal aproximação é realizada pela tendência assintótica dos conjuntos de detecções com baixa dispersão.

Também foram apresentadas informações sobre a análise dos modelos de predição de séries temporais. Foram avaliados os modelos a seguir: Auto-Regressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR) e Rede Neural Recorrente.

A análise preditiva é uma técnica muito utilizada em aplicações de monitoramento, por isso, os modelos de predição de séries temporais serão aplicados nos conjuntos de dados reais e reconstruídos a fim de constatar o quanto a reconstrução dos dados afeta a predição, bem como, qual modelo de predição se ajusta melhor aos dados. No próximo capítulo serão apresentados os resultados obtidos nos experimentos realizados com relação à abordagem proposta neste trabalho.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo serão apresentados os resultados e a análise de desempenho da abordagem proposta para redução e reconstrução de dados em Redes de Sensores sem Fio (RSSF). Este capítulo será dividido em cinco seções. Na Seção 5.1 são apresentados detalhes sobre os conjuntos de dados (*datasets*) utilizados nos experimentos, bem como o ajuste dos parâmetros N_{max} e e_{max} em cada *dataset*. Na Seção 5.2 descrevem-se os resultados obtidos na redução e reconstrução dos *datasets*. Na Seção 5.3, são apresentados os resultados obtidos com os modelos de predição de séries temporais (ARIMA, SVM e Redes Neurais Recorrentes), avaliando a previsão de valores futuros com base nos dados reais e nos dados reconstruídos pela abordagem proposta. Na Seção 5.4 descrevem-se os resultados obtidos nas simulações com o Simulador Castalia (RASTEGARNIA; SOLOUK, 2011), com o intuito de avaliar o impacto gerado pelo uso da abordagem proposta com relação à redução das transmissões de dados, reconstrução de dados, consumo de energia e atraso na entrega das mensagens. Por fim, na Seção 5.5 são apresentadas as conclusões do capítulo.

5.1 CONJUNTOS DE DADOS AVALIADOS

Nos experimentos da abordagem proposta neste trabalho, foram utilizados oito conjuntos de dados (*datasets*) reais de aplicações que realizam monitoramento através de sensores. São eles:

1. Monitoramento de compressores de ar de cadeiras odontológicas (CASTAÑEDA, 2016);

Cadeiras odontológicas contém compressores de ar. Frequentemente, parte do ar comprimido nos compressores sofre condensação, e consequentemente, os compressores ficam cheios de água, tornando as cadeiras inoperantes. Para evitar que as cadeiras fiquem fora de operação, sensores foram utilizados para monitorar a umidade nos compressores de cadeiras odontológicas a fim de expurgar a água acumulada em momento oportuno.

2. Dados do medidor inteligente de energia elétrica (*SmartMeter*) do Projeto *Low Carbon London* - UKPN (NETWORKS, 2015);

O projeto *Low Carbon London* disponibiliza dados de medidores inteligentes de 5.567 lares participantes em Londres - Reino Unido, no período de Novembro de 2011 a Fevereiro de 2015.

3. Dados ambientais de monitoramento da qualidade da água do Rio Burnett - Governo do Queensland (Austrália) (QUEENSLAND GOVERNMENT, 2015);

Este *dataset* coleta séries temporais de leituras de parâmetros sobre a qualidade da água estuariana a cada meia hora no período de 2016 a 2018.

4. Dados de sensores de qualidade da água da praia *Ohio Street Beach* - Chicago Data Portal (CITY OF CHICAGO, 2019a);

Neste *dataset*, são apresentadas leituras referentes à temperatura da água, turbidez e altura de onda. Estes dados são disponibilizados pelo Governo de Chicago, no site Chicago Data Portal.

5. Dados de temperatura capturados no Laboratório de Pesquisa da Intel na Universidade da Califórnia em Berkeley (MADDEN, 2004);

Este conjunto de dados provém de 54 sensores Mica2Dot que monitoram os seguintes parâmetros: temperatura, umidade, luminosidade e tensão.

6. Dados de sensores de qualidade do ar - Universidade da Califórnia em Irvine (VITO et al., 2008);

Este *dataset* está disponível no sítio da *UCI Machine Learning Repository* que contém uma coleção de bases de dados disponibilizadas pela Universidade da Califórnia em Irvine para a comunidade de *Machine Learning*.

7. Dados de sensores de qualidade do ar localizados em praias - Chicago Data Portal (CITY OF CHICAGO, 2019b);

Leituras referentes à umidade, direção do vento, velocidade do vento e pressão barométrica são reunidos neste *dataset* disponibilizado pelo Governo de Chicago, no site Chicago Data Portal.

8. Dados de nós sensores *Tmote Sky* utilizados no Departamento de Ciência da Computação da ULB (Grupo de Machine Learning) (UNIVERSITÉ LIBRE DE BRUXELLES, 2012);

Este *dataset* contém dados de temperatura, umidade e luminosidade coletados por 20 nós sensores *Tmote Sky* no Laboratório de Redes de Sensores sem Fio do Departamento de Ciência da Computação da Université Libre de Bruxelles.

Na Tabela 5.1 são apresentados detalhes sobre os *datasets* utilizados nos experimentos, bem como as suas classificações de acordo com a taxonomia descrita no Capítulo 4, Seção 4.3. Os *datasets* são divididos em classes (0, 1 e 2) e os parâmetros N_{max} e e_{max} foram ajustados consoante aos seus requisitos.

Tabela 5.1 – Detalhes dos Datasets Utilizados

#	Parâmetros	Amostragem	Período	Nº de Amostras	Fonte	Classe	N_{max}	e_{max}
#1.1	Temperatura	$\pm 2s$	± 1 ano	3.970.215	(CASTAÑEDA, 2016)	0	15	0,25
#1.2	Umidade	$\pm 2s$	± 1 ano	3.970.215	(CASTAÑEDA, 2016)	0	15	0,25
#1.3	Ponto de Condensação	$\pm 2s$	± 1 ano	3.970.215	(CASTAÑEDA, 2016)	0	15	0,25
#2.1	kWh/ hh	30min	1 ano	17.458	(NETWORKS, 2015)	0	15	0,25
#3.1	pH	30min	1 ano	14.332	(QUEENSLAND GOVERNMENT, 2015)	1	30	0,35
#3.2	Oxigênio Dissolvido (mg/l)	30min	1 ano	14.332	(QUEENSLAND GOVERNMENT, 2015)	1	30	0,35
#4.1	Temperatura da Água	1 hora	2 anos	11.292	(CITY OF CHICAGO, 2019a)	1	30	0,35
#4.2	Turbidez	1 hora	2 anos	11.292	(CITY OF CHICAGO, 2019a)	1	30	0,35
#4.3	Altura de Onda	1 hora	2 anos	11.292	(CITY OF CHICAGO, 2019a)	1	30	0,35
#5.1	Temperatura	$\pm 30s$	4 dias	43.046	(MADDEN, 2004)	2	45	0,50
#5.2	Umidade	$\pm 30s$	4 dias	43.046	(MADDEN, 2004)	2	45	0,50
#5.3	Luminosidade	$\pm 30s$	4 dias	43.046	(MADDEN, 2004)	2	45	0,50
#5.4	Voltagem	$\pm 30s$	4 dias	43.046	(MADDEN, 2004)	2	45	0,50
#6.1	Temperatura	1 hora	1 ano	9.357	(VITO et al., 2008)	2	45	0,50
#6.2	Umidade Relativa	1 hora	1 ano	9.357	(VITO et al., 2008)	2	45	0,50
#6.3	Umidade Absoluta	1 hora	1 ano	9.357	(VITO et al., 2008)	2	45	0,50
#7.1	Umidade	± 1 hora	7 meses	24.716	(CITY OF CHICAGO, 2019b)	2	45	0,50
#7.2	Direção do Vento	± 1 hora	7 meses	24.716	(CITY OF CHICAGO, 2019b)	2	45	0,50
#7.3	Velocidade do Vento	± 1 hora	7 meses	24.716	(CITY OF CHICAGO, 2019b)	2	45	0,50
#7.4	Temperatura de Bulbo Úmido	± 1 hora	7 meses	24.716	(CITY OF CHICAGO, 2019b)	2	45	0,50
#8.1	Temperatura	20min	5 dias	1.195	(UNIVERSITÉ LIBRE DE BRUXELLES, 2012)	2	45	0,50
#8.2	Umidade	20min	5 dias	1.195	(UNIVERSITÉ LIBRE DE BRUXELLES, 2012)	2	45	0,50
#8.3	Luminosidade	20min	5 dias	1.195	(UNIVERSITÉ LIBRE DE BRUXELLES, 2012)	2	45	0,50

5.2 ANÁLISE DA REDUÇÃO E RECONSTRUÇÃO DOS DADOS

Para avaliar o potencial dos algoritmos de redução e reconstrução de dados descritos no Capítulo 4, foram realizados experimentos a fim de identificar o percentual de redução alcançado em cada *dataset*, bem como, o erro obtido nos dados finais. Para fins de comparação com outros trabalhos da literatura que lidam com o mesmo tema, foram implementadas três abordagens cujos detalhes serão apresentados a seguir:

- *Adaptive Method for Data Reduction* (AMDR):

No trabalho de Fathy, Barnaghi e Tafazolli (2018) foi proposta uma abordagem adaptativa para redução das transmissões de dados em Redes de Sensores sem Fio (RSSF) baseada no algoritmo *Least Mean Square* (LMS) e utiliza uma combinação convexa de dois filtros com diferentes tamanhos para estimar detecções futuras com maior acurácia.

- *Water Meter Readings Prediction* (WMRP):

Almhana, Choulakian e Almhana (2017) propuseram uma abordagem para redução de dados em RSSF baseada no modelo estatístico de predição ARIMA somado a estratégias de análise de dispersão para alcançar maior redução na transmissão de dados a partir de sensores de consumo de água.

- *Quality-Based Data Reduction* (QBDR):

No trabalho de Santini e Romer (2006) foi proposta uma abordagem para redução das transmissões de dados em RSSF com base em um esquema de predição dupla baseado no algoritmo *Least Mean Square* (LMS) com um filtro adaptativo que realiza a previsão de detecções futuras.

As três abordagens escolhidas para comparação são baseadas em técnicas de predição de dados. Foram escolhidos trabalhos baseados em técnicas de predição pelo fato de haver uma certa similaridade com a abordagem proposta, no sentido de realizar a redução no nó sensor. Além disso, alguns dos trabalhos escolhidos para comparação (FATHY; BARNAGHI; TAFAZOLLI, 2018; SANTINI; ROMER, 2006) avaliaram em seus experimentos, o *dataset #5* (MADDEN, 2004), apresentado na Tabela 5.1, também utilizado neste trabalho a fim de obter uma comparação ainda mais relevante e justa.

A implementação da abordagem proposta e dos demais trabalhos relacionados (AMDR, WMRP e QBDR) foi realizada na linguagem de programação C++, em uma máquina com as seguintes configurações: Processador Intel Core i5-4590S CPU 3.00GHz de 4 núcleos e 8GB de memória RAM.

Nos *datasets* utilizados, a quantidade de sensores inclusos varia, bem como a frequência de captura dos dados. Por exemplo, o *dataset* de monitoramento de compressores (#1.1, #1.2 e #1.3) contém dados de apenas um sensor, e a frequência de captura dos dados é de 2 segundos, aproximadamente. Já o *dataset* do Laboratório *Intel Berkeley Research* (#5.1, #5.2, #5.3 e #5.4) contém dados de 54 sensores, e a frequência de captura dos dados é de 30 segundos aproximadamente. No experimento apresentado na Tabela 5.2, apenas os dados de um sensor proveniente de cada *dataset* é apresentado na Tabela 5.1.

Na Tabela 5.2 são apresentados os resultados dos experimentos realizados com a abordagem proposta e com os trabalhos AMDR, WMRP e QBDR. As métricas avaliadas são: quantidade de transmissões suprimidas e o Erro Absoluto Médio (EAM) obtido nos dados finais, que representa a acurácia do algoritmo de reconstrução (Algoritmo 4.1).

Através dos resultados apresentados na Tabela 5.2, verificou-se que o comportamento das séries temporais são determinantes para o processo de redução de dados. Séries temporais com valores muito dispersos geralmente alcançam menor redução, como é o caso do *dataset* de consumo de energia do projeto *Low Carbon London* (#2), no qual foi alcançado apenas 10,65% de redução com a abordagem proposta. Com o mesmo *dataset*, o método *Water Meter Readings Prediction* (WMRP) obteve 5,38% de redução nas transmissões. Já os métodos *Adaptive Method for Data Reduction* (AMDR) e *Quality-Based Data Reduction* (QBDR) não foram capazes de reduzir as transmissões (0% de redução).

É importante salientar que, de acordo com os resultados apresentados, os métodos *Adaptive Method for Data Reduction* (AMDR) e *Quality-Based Data Reduction* (QBDR) demonstram maiores limitações em séries temporais mais dispersas, como é o caso das séries temporais de umidade do *dataset* (#5.2) e velocidade do vento do *dataset* de sensores de qualidade do ar (#7.3). Nestes casos, nenhuma redução foi alcançada. Com os mesmos *datasets*, a abordagem proposta, atingiu um percentual de redução de 73,27% e 5,64%, respectivamente.

Por outro lado, nos *datasets* de monitoramento da temperatura interna de compressores de ar (#1.1), pH da água (#3.1) e voltagem (#5.4), os resultados obtidos pela abordagem proposta foram inferiores àqueles obtidos pelo método *Adaptive Method for Data Reduction* (AMDR), bem como pelo método *Water Meter Readings Prediction* (WMRP). Nesses *datasets*, o comportamento padrão da série temporal é de estacionariedade. Ou seja, os valores detectados ficam ao redor de uma média constante, refletindo uma determinada estabilidade. Nesses casos, a qualidade dos valores previstos pelos métodos AMDR e WMRP aumenta.

Tabela 5.2 – Análise da Redução e Reconstrução de Dados (EAM)

#	Dataset	Abordagem Proposta	AMDR (FATHY; BARNAGHI; TAFAZOLLI, 2018)	WMRP (ALMHANA; CHOUAKIAN; ALMHANA, 2017)	QBDR (SANTINI; ROMER, 2006)								
#	Dataset	Redução (nº)	EAM	Redução (nº)	EAM	Redução (nº)	EAM	Redução (%)	EAM	Redução (nº)	EAM	Redução (%)	EAM
#1.1	Temperatura	3652520	91,99	0,017	3678407	92,65	6,54E-16	3586205	90,33	0,000669	727163	18,31	0,028
#1.2	Umidade	3642755	91,75	0,024	3613420	91,013	1,89E-16	3627188	91,36	0,000453	155269	3,91	0,0047
#1.3	Ponto de Condensação	3645919	91,83	0,022	3599495	90,662	0,019972	3627188	91,36	0,00271	1160732	29,23	0,0344
#2.1	kWh/hh	1860	10,65	0,017	0	0	0	940	5,384	0,0012	0	0	0
#3.1	pH	13864	96,73	0,043	14109	98,444	0,097088	14265	99,53	0,0151	11632	81,16	0,0723
#3.2	Oxigênio Dissolvido	13846	96,6	0,24	9845	68,692	0,092164	13731	95,8	0,072	8139	56,78	0,0792
#4.1	Temperatura	9650	85,47	0,385	2676	23,702	0,035266	7886	69,84	0,0805	1442	12,77	0,02
#4.2	Turbidez	9845	87,2	0,332	3282	29,07	0,038891	7626	67,54	0,0786	3	0,02	6,43E-05
#4.3	Altura de Onda	10899	96,53	0,031	3349	29,663	0,033498	11181	99,03	0,0188	3035	26,88	0,0392
#5.1	Temperatura	28139	65,36	0,021	16	0,03717	2,62E-05	23744	55,15	0,0149	0	0	0
#5.2	Umidade	31541	73,27	0,07	0	0	0	30492	70,83	0,0454	0	0	0
#5.3	Luminosidade	30846	71,65	0,026	15590	36,217	0,00339962	30443	70,72	0,0097	159	0,36	9,5926
#5.4	Voltagem	42105	97,81	0,049	42978	99,842	0,13328	42980	99,84	0,0522	189	0,43	4,7347
#6.1	Temperatura	6504	69,5	0,451	600	6,4123	0,0074648	3518	37,59	0,0832	102	1,09	0,0021
#6.2	Umidade Relativa	4780	51,08	0,344	293	3,1313	3,53E-05	1302	13,91	0,0285	2	0,02	7,9024
#6.3	Umidade Absoluta	9135	97,62	0,097	8921	95,34	0,1449	9274	99,11	0,0309	2126	22,72	0,0841
#7.1	Umidade	21172	85,66	0,869	2847	11,519	0,024588	13765	55,69	0,1178	1744	7,05	0,0158
#7.2	Direção do Vento	7455	30,16	0,071	16	0,064735	2,30E-18	2803	11,34	0,0006	5	0,02	7,7505
#7.3	Velocidade do Vento	1394	5,64	0,002	0	0	0	546	2,209	0,0004	0	0	0
#7.4	Temp. de bulbo úmido	21067	85,23	0,594	899	3,6373	0,0088037	10046	40,64	0,0942	760	3,07	0,0072
#8.1	Temperatura	1168	97,74	0,123	1066	89,205	0,15382	1142	95,56	0,0858	887	74,22	0,1633
#8.2	Umidade	1169	97,82	0,111	1116	93,389	0,14624	233	82,33	0,0781	850	71,13	0,1112
#8.3	Luminosidade	1169	97,82	0,057	1131	94,644	0,21011	1145	95,81	0,0331	978	81,84	0,1499

A abordagem proposta, por outro lado, possui um fator limitante que é o parâmetro N_{max} . Este parâmetro é utilizado para que os requisitos de cada aplicação de monitoramento com relação à frequência de transmissão de dados sejam respeitados. Por isso a abordagem proposta atingiu resultados inferiores àqueles alcançados pelos métodos *Adaptive Method for Data Reduction* (AMDR) e *Water Meter Readings Prediction* (WMRP) nos *datasets* de temperatura (#1.1), pH (#3.1) e voltagem (#5.4).

Contudo, a abordagem proposta alcança melhores resultados tanto em séries temporais menos dispersas quanto mais dispersas, mas cujo comportamento demonstra sazonalidade. É o caso dos seguintes *datasets*: oxigênio dissolvido (#3.2), temperatura (#4.1), altura de onda (#4.3), temperatura (#5.1), umidade (#5.2), luminosidade (#5.3), temperatura #(6.1), umidade relativa (#6.2), umidade (#7.1), direção do vento (#7.2), velocidade do vento (#7.3), temperatura de bulbo úmido (#7.4), temperatura (#8.1), umidade (#8.2) e luminosidade (#8.3). Em todos estes casos, à medida em que os métodos AMDR, WMRP e QBDR perdem qualidade, a abordagem proposta se beneficia de pequenos intervalos de baixa dispersão alcançando resultados superiores.

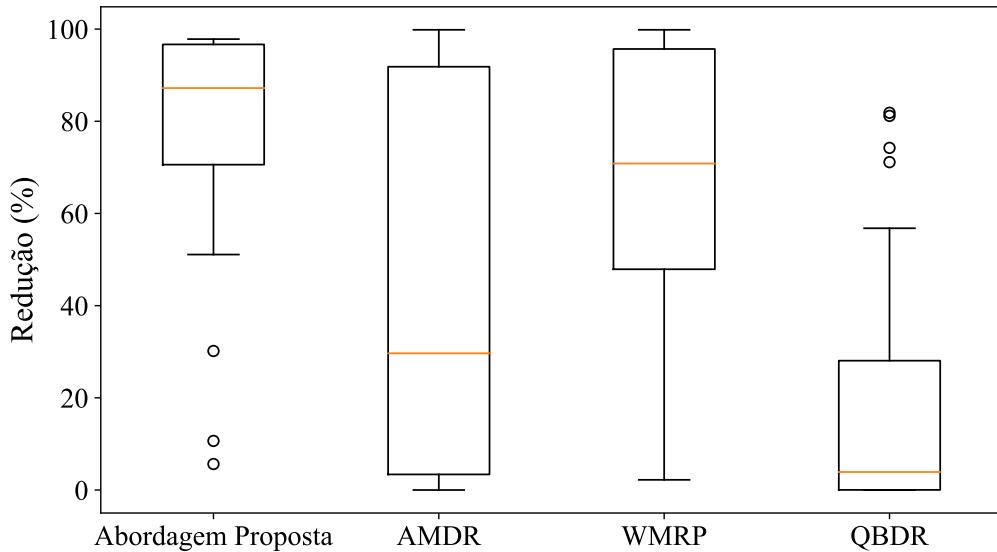
De modo geral, as séries temporais menos dispersas alcançam maior nível de redução, como por exemplo, o *dataset* da Universidade Livre de Bruxelas (#8.1, #8.2 e #8.3), no qual todos os métodos alcançaram redução superior à 70%. Outro exemplo é o *dataset* de oxigênio dissolvido (#3.2), no qual a abordagem proposta, bem como as demais técnicas (AMDR, WMRP e QBDR) atingiram alto nível de redução.

A fim de obter melhor visualização dos resultados apresentados na Tabela 5.2, são ilustrados, nas Figuras 5.1 e 5.2, diagramas de caixa que demonstram a variação dos resultados obtidos com relação à porcentagem de redução e Erro Absoluto Médio (EAM) em cada método de redução de dados, por meio de quartis. Estes diagramas são construídos com base nos resultados de cada método de redução de dados (Abordagem proposta, AMDR, WMRP e QBDR) em todos os *datasets* apresentados na Tabela 5.1.

De acordo com os resultados apresentados na Figura 5.1, pode-se constatar que a abordagem proposta, de forma geral, alcança melhores resultados com relação à redução das transmissões de dados. Especificamente no caso da abordagem proposta, pode-se perceber a existência de valores atípicos (discrepantes) representados com pontos individuais. Tais valores se referem aos poucos casos em que a abordagem proposta teve um baixo desempenho se comparado aos demais, pois os pontos individuais estão posicionados abaixo da cauda inferior.

Observa-se também a existência de valores discrepantes nos resultados do método *Quality-Based Data Reduction* (QBDR). Nesse caso, os valores discrepantes

Figura 5.1 – Percentual de Redução na Transmissão de Dados. Fonte: Próprio Autor



estão posicionados acima da cauda superior e, consequentemente, representam os poucos casos em que o método QBDR obteve resultados mais significativos.

Com relação aos métodos *Adaptive Method for Data Reduction* (AMDR) e *Water Meter Readings Prediction* (WMRP), a variação dos resultados se deu de forma similar, porém a mediana é mais elevada no caso do método WMRP, demonstrando maior desempenho se comparado ao AMDR. Já o método *Quality-Based Data Reduction* (QBDR) obteve o pior desempenho, em comparação aos demais métodos.

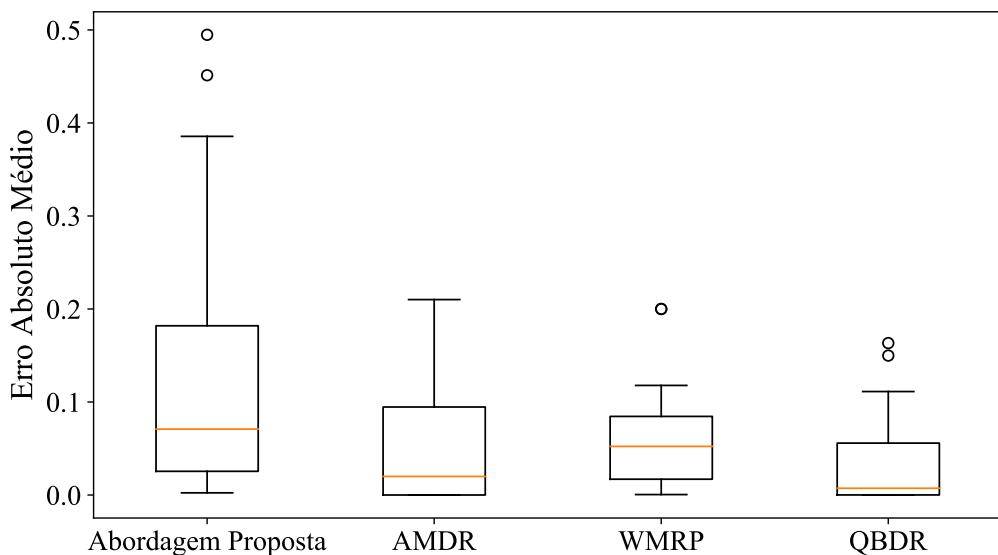
Outro ponto importante a ser analisado, é o Erro Absoluto Médio (EAM) obtido nos dados finais em cada técnica de redução de dados. No caso da abordagem proposta, os dados finais são os dados gerados pelo algoritmo de reconstrução. Logo, através do Erro Absoluto Médio (EAM) pode-se avaliar a acurácia da reconstrução.

Na Figura 5.2 são ilustrados diagramas de caixa que apresentam a distribuição dos resultados com relação ao Erro Absoluto Médio (EAM) obtido por cada método de redução de dados em todos os *datasets* apresentados na Tabela 5.1.

É importante enfatizar que, nos métodos utilizados para redução de dados, existe um *trade-off* entre percentual de redução a ser alcançada e acurácia nos dados finais. Logo, quanto maior o nível de redução, maior será o erro obtido. Do mesmo modo, quanto menor for a redução, menor será o erro obtido.

Os resultados apresentados na Figura 5.2 mostram que a abordagem proposta obteve o maior Erro Absoluto Médio (EAM) em contraste com as demais técnicas utilizadas (AMDR, WMRP e QBDR). Isto acontece por conta da proporcionalidade existente entre a redução das transmissões de dados e a acurácia dos dados finais,

Figura 5.2 – Erro Absoluto Médio (EAM). Fonte: Próprio Autor



uma vez que a abordagem proposta alcançou maior nível de redução de forma geral. Ainda assim, em nenhum dos *datasets* o erro obtido foi maior que o máximo aceitável (e_{max}), calculado com base nos requisitos de cada aplicação, conforme descrito na Tabela 5.1 (última coluna). Até mesmo os valores discrepantes presentes no diagrama da Figura 5.2 são inferiores a 0,5, que é o valor de erro máximo aceitável para os *datasets* de classe 2, de acordo com taxonomia apresentada no Capítulo 4.

Os métodos AMDR e QBDR, porém, obtiveram menor erro (maior acurácia) pelo fato de terem alcançado menor redução nas transmissões de dados.

Com a finalidade de avaliar a acurácia dos resultados obtidos com a abordagem proposta, bem como com os métodos *Adaptive Method for Data Reduction* (AMDR), *Water Meter Readings Prediction* (WMRP) e *Quality-Based Data Reduction* (QBDR), foram utilizados testes estatísticos que indicam a existência de significância estatística nos resultados. Detalhes sobre os testes utilizados serão abordados a seguir.

5.2.1 Testes de Significância Estatística

A fim de avaliar estatisticamente os resultados apresentados na Tabela 5.2, os seguintes testes estatísticos foram utilizados: Shapiro-Wilk (RAZALI; WAH et al., 2011), Teste t de Student (RAJU, 2005) e Wilcoxon (ROSNER; GLYNN; LEE, 2006).

O teste de Shapiro-Wilk testa a hipótese nula de que os dados de entrada foram extraídos de uma distribuição normal. Caso os dados de entrada, neste caso representados pelos resultados obtidos com as técnicas de redução de dados, sigam

uma distribuição normal, o Teste t de Student é aplicado. Caso os dados de entrada não sigam uma distribuição normal, o teste Wilcoxon é aplicado.

O Teste t de Student é um teste paramétrico, no qual os dados de entrada precisam seguir uma distribuição normal. Já o teste Wilcoxon se trata de um teste estatístico não-paramétrico no qual os dados de entrada não seguem uma distribuição normal. Como resultado, os testes indicam se os dados de entrada se diferem estatisticamente ou não, baseando-se em um intervalo de confiança de 95%. Dessa forma, se um conjunto de dados seguir uma distribuição normal, assume-se que o intervalo de confiança inclui a média dos dados com 95% de confiança.

Os testes estatísticos foram executados com base nos resultados apresentados na Tabela 5.2. Foi identificado que os dados não seguem uma distribuição normal, e por isso o teste Wilcoxon foi utilizado. A média dos resultados obtidos com relação à redução de dados, pela abordagem proposta e pelos métodos AMDR, WRMP e QBDR em todos os *datasets* foi de 77,17%, 45,97%, 66,99% e 21,34% respectivamente. Os testes estatísticos demonstram que há diferença estatística nos resultados obtidos, confirmando que a abordagem proposta obteve o melhor desempenho no âmbito geral.

Na próxima seção, modelos de previsão de séries temporais serão avaliados, com o intuito de verificar se a reconstrução dos dados afeta de alguma forma a previsão de valores futuros.

5.3 AVALIAÇÃO DA PREDIÇÃO DE VALORES FUTUROS

Nos experimentos apresentados nesta seção, avalia-se a acurácia dos dados resultantes da redução, em relação à previsão de valores futuros. Para realizar esta análise, foram utilizados os seguintes modelos de predição encontrados na literatura: *Auto-Regressive Integrated Moving Average* (ARIMA), *Support Vector Machine* (SVM) e Redes Neurais Recorrentes. Os modelos de predição receberam como entrada cada um dos *datasets* (reais e reconstruídos) em cada execução. O objetivo principal é verificar se o processo de redução e reconstrução dos dados através do método proposto neste trabalho afeta de alguma forma a previsão de valores futuros.

Especificamente, foram utilizadas apenas as últimas 10.000 detecções do *dataset* de monitoramento de compressores de ar (#1) para a previsão de valores futuros, pelo fato de ser um *dataset* consideravelmente extenso (quase quatro milhões de detecções). Os demais *datasets* foram utilizados em sua totalidade.

A implementação do algoritmo de predição baseado no modelo *Auto-Regressive Integrated Moving Average* (ARIMA) foi realizada na linguagem de programação *Python*,

juntamente com a biblioteca *statsmodels*¹ que fornece classes e funções para a estimativa de diferentes modelos estatísticos, bem como para a realização de testes estatísticos e exploração de dados estatísticos.

A implementação do algoritmo de predição baseado no modelo SVR, também foi realizada na linguagem de programação *Python*, juntamente com a biblioteca *scikit-learn*² que fornece vários algoritmos de classificação, regressão e agrupamento e é projetada para interagir com as bibliotecas numéricas e científicas da linguagem *Python*, como NumPy³ e SciPy⁴.

Com relação à Rede Neural Recorrente, foram definidos 75% dos dados para o conjunto de treino e 25% para o conjunto de teste. A implementação da Rede Neural se deu através do uso da ferramenta TensorFlow (ABADI et al., 2016). O TensorFlow é um projeto do Google focado na implementação e implantação de modelos de Aprendizagem de Máquina em larga escala. Sua abordagem é baseada em um modelo flexível de programação baseada em fluxo de dados aplicado a diversas plataformas de hardware, sejam elas centralizadas ou distribuídas. É possível utilizá-lo ainda para expressar uma grande variedade de algoritmos, incluindo algoritmos de treinamento para Redes Neurais de forma escalável. A Interface de Programação de Aplicativos (API) principal é na linguagem de programação Python, com implementação de computações extremamente eficientes também em C++ (ABADI et al., 2016).

Ressalta-se que outros modelos além do TensorFlow podem ser utilizados, como Torch (COLLOBERT; FARABET; KAVUKCUOĞLU, 2008), Theano (BASTIEN et al., 2012) e Neupy (SHARP, 2016). O TensorFlow foi escolhido dada a sua eficiência nos resultados e facilidade de implementação. Além do TensorFlow, foram utilizadas outras bibliotecas existentes para a manipulação dos dados, como Pandas⁵, NumPy e Matplotlib⁶.

Alguns parâmetros da Rede Neural precisaram ser ajustados, como: número de neurônios, taxa de aprendizado, e tamanho dos mini-lotes. Tais parâmetros foram ajustados de forma *offline*, ou seja, antes da execução do algoritmo. A escolha dos valores desses parâmetros foi realizada de forma empírica, baseando-se na observação do comportamento do algoritmo. São eles: Número de Neurônios: 64; taxa de aprendizado: 0,001 e por fim, Tamanho dos mini-lotes: 64.

Para cada *dataset* (real e reconstruído), o modelo baseado na Rede Neural foi executado 10 vezes para cada *dataset*. Após as 10 execuções, foram calculados a mé-

¹ <https://www.statsmodels.org/stable/index.html>

² <http://scikit-learn.org/stable/index.html>

³ <http://www.numpy.org/>

⁴ <https://www.scipy.org/>

⁵ <https://pandas.pydata.org/>

⁶ <https://matplotlib.org/>

dia e Desvio Padrão (DP) do Erro Absoluto Médio (EAM), Coeficiente de Determinação (R^2), Tempo de execução (TE) e quantidade de iterações no conjunto (*set*) de treino. A quantidade de execuções não foi maior devido ao tempo de execução que pode se tornar demasiadamente alto, dificultando a obtenção dos resultados. Os demais algoritmos (ARIMA e SVM) foram executados apenas uma vez para cada *dataset*, uma vez que não se tratam de modelos estocásticos.

Os experimentos com os algoritmos *Auto-Regressive Integrated Moving Average* (ARIMA) e *Support Vector Machine* (SVM) e Rede Neural Recorrente foram executados em uma máquina virtual na plataforma Azure⁷, equipada com um processador de 4 núcleos e 8GB de memória RAM. No caso da Rede Neural Recorrente, a etapa de treinamento foi paralelizada através do TensorFlow, ocorrendo em todos os núcleos do processador. Optou-se por realizar estes experimentos na plataforma Azure pelo longo tempo de processamento da Rede Neural Recorrente. Dessa forma, a máquina virtual foi dedicada exclusivamente à estes experimentos.

Durante a execução dos experimentos, foram monitorados os seguintes quesitos: Erro Absoluto Médio (EAM), Coeficiente de Determinação (R^2), Tempo de execução (T.E) e quantidade de iterações no *set* de treino (apenas para a RNN). Na Rede Neural Recorrente, o número de iterações no *set* de treino pode variar, pelo fato de ser ajustado de forma automática durante a execução do algoritmo. É importante ressaltar que determinadas séries temporais podem demonstrar comportamentos que exijam maior quantidade de iterações de treinamento para que a otimização do Erro Absoluto Médio (EAM) chegue o mais próximo possível do ótimo.

Os resultados obtidos na previsão de valores futuros estão descritos na Tabela 5.3. A fim de obter melhor visualização, os resultados são ilustrados nas Figuras 5.3, 5.4 e 5.5 através de diagramas de caixa.

Os resultados apresentados na Tabela 5.3 demonstram que os valores obtidos com relação às métricas de Erro Absoluto Médio (EAM) e R^2 não diferem muito entre os *datasets* reais e *datasets* reconstruídos. De maneira geral, os resultados de Erro Absoluto Médio (EAM) obtidos são baixos (entre 0 e 1), com algumas exceções, como é o caso dos seguintes *datasets*: consumo de energia (#2.1), altura de onda (#4.3), temperatura (#5.1), umidade (#5.2), luminosidade (#5.3) e velocidade do vento (#7.3), cujo nível de erro obtido foi maior, devido ao comportamento de dispersão destas séries que dificulta a previsão de valores futuros.

Especificamente no *dataset* de temperatura (#5.1), os modelos de predição RNN, ARIMA e SVM obtiveram um Erro Absoluto Médio (EAM) de 23922,46, 27272,57 e 19597,04 respectivamente, representando uma discrepância com relação aos de-

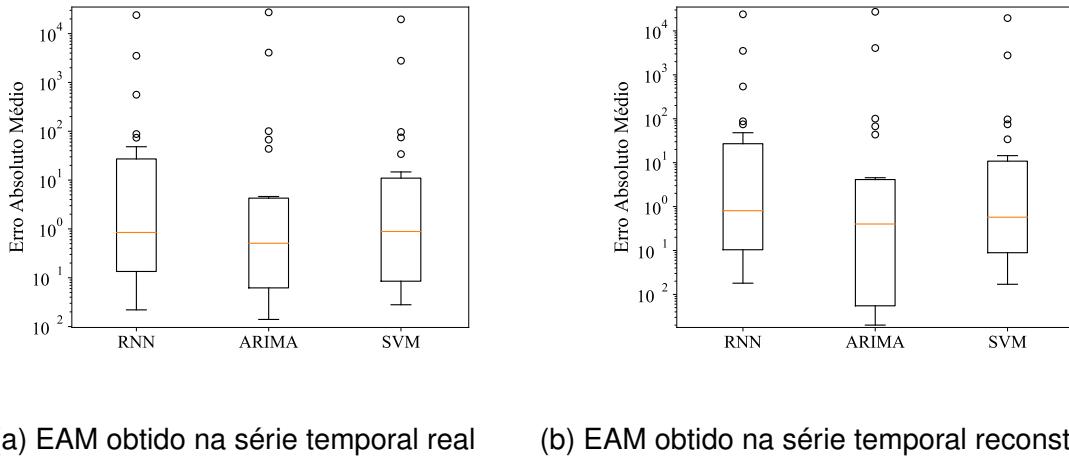
⁷ <https://azure.microsoft.com/pt-br/>

Tabela 5.3 – Análise da Predição de Valores Futuros

Datasets	REDE NEURAL RECORRENTE (RNN)								ARIMA			SVM		
	EAM		R ²		T.E.(s)		Iterações		EAM	R ²	T.E. (s)	EAM	R ²	T.E. (s)
	Média	D.P.	Média	D.P.	Média	D.P.	Média	D.P.						
#1.1 (Real)	0,327	0,174	0,936	0,054	1416,54	687,511	2619,2	743,17	0,024	0,993	15,746	0,04	0,991	5,119
#1.1 (Reconstruído)	0,245	0,094	0,968	0,019	1701,35	935,751	3688,4	1136,621	0,004	0,998	28,403	0,029	0,993	4,85
#1.2 (Real)	0,106	0,015	0,676	0,031	1233,845	607,855	2366,1	412,379	0,023	-0,155	13,639	0,028	0,348	4,422
#1.2 (Reconstruído)	0,096	0,027	0,864	0,052	1241,103	664,096	2641,6	694,194	0,003	0,934	26,725	0,017	0,675	3,2592
#1.3 (Real)	0,131	0,019	0,794	0,037	1373,515	809,124	2894,1	1191,24	0,018	0,917	16,018	0,028	0,946	5,42
#1.3 (Reconstruído)	0,107	0,014	0,868	0,056	1330,516	682,431	2865,0	726,99	0,003	0,994	29,441	0,018	0,97	3,485
#2.1 (Real)	87,736	0,94	0,107	0,01	9280,9	4915,078	10000,0	0,0	100,498	-0,342	20,283	96,748	-0,031	4,767
#2.1 (Reconstruído)	87,48	0,878	0,106	0,012	7916,544	4104,696	10000,0	0,0	100,491	-0,342	58,115	96,748	-0,031	5,333
#3.1 (Real)	0,022	0,007	0,977	0,01	7420,773	3898,459	9718,1	767,517	0,018	0,973	26,207	0,062	0,816	7,681
#3.1 (Reconstruído)	0,018	0,009	0,988	0,011	4487,797	2563,846	7491,2	2270,589	0,004	0,999	51,643	0,051	0,882	1,3705
#3.2 (Real)	0,103	0,0	0,95	0,001	7709,131	4054,817	9780,2	472,771	0,147	0,875	23,796	0,296	0,656	59,338
#3.2 (Reconstruído)	0,028	0,004	0,972	0,001	5466,636	2843,023	8505,1	1366,885	0,035	0,927	49,26	0,118	0,858	16,9
#4.1 (Real)	0,717	0,009	0,675	0,004	6212,338	3211,22	10000,0	0,0	0,508	0,679	12,761	0,887	0,605	30,073
#4.1 (Reconstruído)	0,615	0,016	0,671	0,003	5270,234	2729,454	10000,0	0,0	0,401	0,673	27,695	0,573	0,655	27,009
#4.2 (Real)	1,115	0,068	-0,675	0,557	6416,605	3329,119	10000,0	0,0	0,334	0,635	11,66	0,457	0,6	39,5622
#4.2 (Reconstruído)	0,942	0,038	-0,498	0,124	5363,636	2759,941	10000,0	0,0	0,175	0,657	27,746	0,294	0,66	23,99
#4.3 (Real)	560,832	72,629	0,165	0,094	5952,293	3084,942	10000,0	0,0	0,014	0,744	10,847	0,038	0,068	0,91157
#4.3 (Reconstruído)	542,688	31,99	0,224	0,055	5391,117	2782,881	10000,0	0,0	0,002	0,984	26,645	0,033	0,087	1,0742
#5.1 (Real)	23922,46	25,724	-0,278	0,004	2209,714	1152,984	10000,0	0,0	27272,577	-0,03	83,877	19597,049	0,277	118,322
#5.1 (Reconstruído)	23928,404	10,202	-0,279	0,001	1853,451	966,095	10000,0	0,0	27272,575	-0,03	348,35	19597,045	0,277	140,08
#5.2 (Real)	3514,333	3,868	-0,077	0,005	2225,962	1161,753	10000,0	0,0	4081,635	-0,614	83,949	2778,858	-0,067	115,779
#5.2 (Reconstruído)	3517,047	1,974	-0,081	0,006	1853,898	965,385	10000,0	0,0	4081,608	-0,614	329,48	2778,845	-0,067	101,91
#5.3 (Real)	48,236	0,37	0,565	0,018	14252,553	7118,851	6272,6	1545,186	43,596	0,427	95,988	34,095	0,584	88,662
#5.3 (Reconstruído)	48,069	0,379	0,573	0,016	9658,143	5899,113	5844,9	1906,778	43,594	0,427	327,82	34,084	0,584	90,82
#5.4 (Real)	0,138	0,003	-0,0	0,0	11858,101	6387,644	5462,9	1347,015	0,451	-1	119,99	0,272	0	259,527
#5.4 (Reconstruído)	0,101	0,016	-0,059	0,177	8940,184	4865,943	4953,9	1426,87	0,383	-1	320,47	0,238	0	16,68
#6.1 (Real)	1,837	0,037	0,914	0,001	5331,936	2812,797	10000,0	0,0	1,514	0,939	9,0156	6,924	0,746	9,27472
#6.1 (Reconstruído)	1,629	0,056	0,916	0,002	4666,936	2401,334	10000,0	0,0	1,356	0,939	19,723	6,614	0,746	13,124
#6.2 (Real)	3,977	0,07	0,913	0,002	5158,814	2676,917	10000,0	0,0	3,932	0,94	8,9495	14,693	0,716	4,42609
#6.2 (Reconstruído)	3,855	0,102	0,913	0,004	4618,117	2406,562	10000,0	0,0	3,706	0,94	20,529	14,456	0,718	5,079
#6.3 (Real)	0,811	0,018	0,916	0,0	1842,554	974,748	3538,8	1128,712	0,704	0,938	10,194	4,237	0,738	7,94678
#6.3 (Reconstruído)	0,805	0,023	0,916	0,0	1404,965	731,32	3055,0	825,232	0,69	0,938	21,525	4,233	0,738	3,414
#7.1 (Real)	1,12	0,008	0,951	0,0	12651,319	6562,462	10000,0	0,0	0,888	0,962	39,967	1,494	0,93	40,428
#7.1 (Reconstruído)	0,805	0,018	0,955	0,0	10693,642	5598,095	10000,0	0,0	0,639	0,965	113,72	1,049	0,945	66,65
#7.2 (Real)	6,004	0,04	0,642	0,005	12188,621	6116,171	10000,0	0,0	4,612	0,771	42,541	7,244	0,583	14,577
#7.2 (Reconstruído)	5,977	0,045	0,641	0,007	10601,545	5527,124	10000,0	0,0	4,546	0,77	116,56	7,195	0,584	15,91
#7.3 (Real)	73,759	0,349	0,155	0,009	7921,638	3441,807	10000,0	0,0	67,281	0,102	34,733	74,689	0,116	6,958
#7.3 (Reconstruído)	74,635	0,705	0,144	0,007	10773,552	5643,328	10000,0	0,0	67,279	0,102	112,56	74,689	0,116	7,647
#7.4 (Real)	0,846	0,006	0,345	0,01	10682,5	5566,334	10000,0	0,0	0,9	0,326	39,369	1,002	0,209	36,456
#7.4 (Reconstruído)	0,453	0,005	0,5	0,002	10702,358	5612,208	10000,0	0,0	0,491	0,507	116,92	0,802	0,284	60,41
#8.1 (Real)	0,105	0,019	0,973	0,008	863,532	534,744	7605,2	3338,539	0,086	0,912	1,343	0,098	0,892	3,181
#8.1 (Reconstruído)	0,11	0,051	0,971	0,023	818,71	401,622	6082,9	3184,954	0,007	0,998	1,6446	0,086	0,924	0,128
#8.2 (Real)	0,229	0,034	0,743	0,066	137,367	67,222	2749,2	808,358	0,093	0,428	0,437	0,118	0,451	0,802
#8.2 (Reconstruído)	0,098	0,03	0,952	0,031	330,994	170,373	7403,6	2130,675	0,01	0,992	0,3609	0,092	0,688	0,278
#8.3 (Real)	0,073	0,026	0,987	0,007	534,287	256,999	4244,7	2343,507	0,038	0,97	1,3329	0,072	0,918	0,087
#8.3 (Reconstruído)	0,098	0,057	0,982	0,014	641,697	314,753	5097,6	1628,396	0,002	0,999	2,3754	0,103	0,858	0,119

mais *datasets*, dado o comportamento da série temporal. Na Figura 5.3, a distribuição dos valores de Erro Absoluto Médio (EAM) apresentados pela análise de todos os *datasets* é representada, utilizando-se a escala logarítmica a fim de proporcionar uma boa visualização dos resultados, dados os valores discrepantes obtidos.

Figura 5.3 – Erro Absoluto Médio (EAM) obtido nos modelos RNN, ARIMA e SVM. Fonte: Próprio Autor



(a) EAM obtido na série temporal real

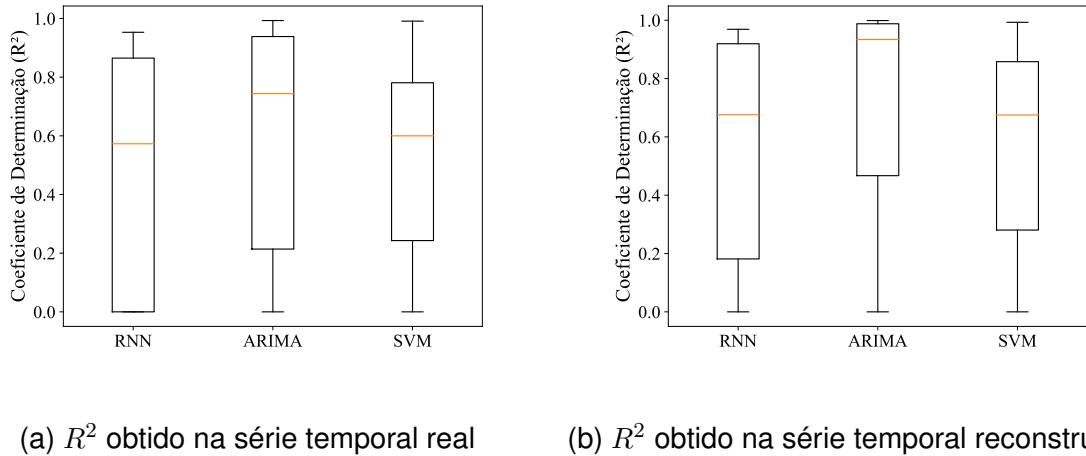
(b) EAM obtido na série temporal reconstruída

A média dos valores obtidos pelos modelos de predição Rede Neural Recorrente, ARIMA e SVM foram 1239,15, 1372,99 e 983,40 respectivamente. Foram realizados testes estatísticos a fim de averiguar a diferença estatística entre os resultados obtidos em cada modelo. Os resultados não seguem uma distribuição normal, e foi identificado, com um intervalo de confiança calculado a 95%, que há diferença estatística nos resultados obtidos pelos modelos de predição. Nesse caso, o modelo SVM obteve o melhor resultado, seguido da Rede Neural Recorrente e do modelo ARIMA, respectivamente.

Outra questão importante é analisar o valor do Coeficiente de Determinação obtido nas séries temporais reais e reconstruídas. Em determinados casos, os valores obtidos foram negativos, demonstrando a baixa qualidade da predição realizada. Por exemplo, no *dataset* de umidade (#5.2), o coeficiente de determinação R^2 foi negativo em todos os modelos de predição. Segundo (IMDADULLAH, 2012), nos casos em que valores negativos de R^2 surgem, a média dos dados reais fornece um melhor ajuste aos resultados do que os valores previstos pelo modelo de predição.

Nos resultados apresentados na Figura 5.4, os valores negativos de coeficiente de determinação obtidos foram alterados para zero, uma vez que qualquer resultado inferior a zero demonstra que a medida de ajustamento não é capaz de explicar os valores previstos. A medida de ajustamento deve variar entre 0 e 1. Assim, quanto mais próximo de 1, melhor a medida de ajustamento explica os valores observados.

Figura 5.4 – R^2 : Modelos RNN, ARIMA e SVM. Fonte: Próprio Autor



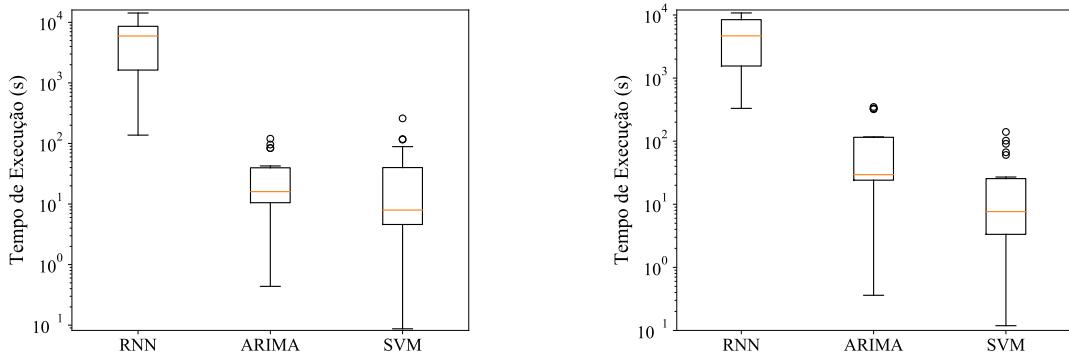
Pode-se identificar, de acordo com os resultados apresentados na Figura 5.4, que os valores obtidos com relação ao Coeficiente de Determinação R^2 não variam muito entre os modelos de predição da Rede Neural Recorrente, ARIMA e SVM, uma vez que a distribuição dos valores observados se dá entre 0 e 1. Especificamente, a média dos valores de R^2 obtidos foram 0,59, 0,58 e 0,52 respectivamente. Os testes estatísticos realizados demonstram que, embora a Rede Neural tenha obtido a melhor média de R^2 , não há diferença estatística entre tais resultados.

Já com relação ao tempo de execução (T.E.) dos modelos de predição, verificou-se que o modelo baseado na Rede Neural Recorrente possui um tempo de execução demasiadamente grande, se comparado aos modelos ARIMA e SVM, conforme apresentado na Figura 5.5. Um exemplo disto é o caso do *dataset* de luminosidade (#5.3), cujo tempo de execução foi em média 14.252,5 segundos (4h aproximadamente). Esta demora se dá pelo fato de que a quantidade de iterações no conjunto de treino depende da minimização do Erro Absoluto Médio (EAM), o que pode levar a um número muito elevado de iterações.

Por fim, foram executados testes estatísticos entre os resultados de Erro Absoluto Médio (EAM) e coeficiente de determinação R^2 resultantes dos modelos de predição utilizando os dados reais, bem como os dados reconstruídos pelo algoritmo de reconstrução proposto. Neste caso, o objetivo é verificar se existe diferença estatística entre os resultados obtidos com os conjuntos de dados reais em comparação com os conjuntos de dados reconstruídos.

No caso da Rede Neural Recorrente, a média de Erro Absoluto Médio (EAM) obtido em todos os *datasets* reais e em todos os *datasets* reconstruídos foi de 1251,12 e 1227,19 respectivamente. No modelo ARIMA, por sua vez, o EAM obtido foi de 1373,03 para os dados reais e 1372,95 para os dados reconstruídos. No modelo SVM,

Figura 5.5 – Tempo de Execução: Modelos RNN, ARIMA e SVM. Fonte: Próprio Autor



(a) Tempo de execução da série temporal real (b) Tempo de execução da série reconstruída

por fim, o EAM obtido nos dados reais e reconstruídos foi de 983,45 e 983,36 respectivamente. Tais resultados não apresentam uma distribuição normal, e foi identificada diferença estatística entre eles, sendo que em todos os modelos avaliados, os valores de EAM são inferiores nos dados reconstruídos se comparados aos valores de EAM obtidos nos dados reais.

Já com relação ao Coeficiente de Determinação R^2 , o modelo de predição Rede Neural Recorrente obteve o R^2 de 0,59 para os dados reais e 0,62 para os dados reconstruídos. Já o modelo ARIMA obteve o R^2 de 0,58 e 0,68 para os dados reais e reconstruídos, respectivamente. O modelo SVM obteve o R^2 de 0,52 para os dados reais e 0,57 para os dados reconstruídos. Embora os resultados não sigam uma distribuição normal, verificou-se que não há diferença estatística entre eles. O intervalo de confiança foi calculado a 95%. Ainda assim, percebe-se que a média de R^2 é superior nos *datasets* reais se comparado aos *datasets* reconstruídos.

Com base nessas informações, verifica-se que a reconstrução dos dados pode impactar na previsão de valores futuros, visto que foi constatada a diferença estatística entre os resultados de Erro Absoluto Médio (EAM) entre os dados reais e reconstruídos. Contudo, destaca-se que os resultados apresentados na Tabela 5.3 em cada *dataset* real e reconstruído, não apresenta alta variabilidade, indicando que o impacto a ser gerado pela reconstrução é consideravelmente baixo.

5.4 RESULTADOS COMPUTACIONAIS NO CASTALIA

Com a finalidade de avaliar a abordagem proposta no contexto de uma Rede de Sensores sem Fio (RSSF), foram realizadas simulações no simulador Castalia⁸

⁸ <https://github.com/boulis/Castalia>

(RASTEGARNIA; SOLOUK, 2011). Nesse sentido, foi reproduzida uma RSSF com 51 nós do *dataset #5.1*, equipados com o rádio CC2420 2.4 GHz IEEE 802.15.4 *ZigBeeReady RF Transceiver*, sendo apenas 1 nó sorvedouro. Os sensores foram distribuídos de forma aleatória em uma distribuição uniforme, através do parâmetro *SN.deployment*, em uma área de 150m x 150m. O sorvedouro, por sua vez, foi posicionado no centro. Os nós da topologia são estáticos.

Com relação ao rádio CC2420, seu raio de alcance é de 50m e a taxa de transmissão é de 250 Kbps. No modo RX, são consumidos 0,062 *joules*/segundo. Já no modo TX são consumidos 0,057 *joules*/segundo e no modo *Idle*, são consumidos 0,014 *joules*/segundo. Nas simulações, a energia inicial dos nós foi definida em 600 *Joules*, sendo que todos os nós sensores iniciam com a mesma quantidade de energia. Estes valores possibilitam que os sensores tenham bateria por 2,45 horas em modo RX, 2,62 horas em modo TX, e 119,04 horas em modo Idle.

O Castalia é um simulador para Redes de Sensores Sem Fio, desenvolvido na linguagem de programação C++ e baseado na plataforma OMNeT++ (VARGA; HORNIG, 2008). Nos experimentos, a versão 3.2 foi utilizada.

O Castalia é dividido em módulos. Dentre eles, destacam-se: Aplicação, Comunicação, Enlace e de camada Física. No módulo de Aplicação, foram desenvolvidas aplicações que contém a implementação da abordagem proposta, bem como dos demais métodos utilizados para comparação (AMDR, WMRP e QBDR). O consumo de energia dessas aplicações foi comparado ao consumo de energia da aplicação *ValueReporting (General)*, que está disponível no Castalia, na qual nenhuma técnica de redução de dados é aplicada. No módulo de Comunicação, utilizou-se o algoritmo de roteamento *Low Energy Adaptive Clustering Hierarchy* (LEACH) (CHEN et al., 2018b). A implementação do LEACH para o Castalia é disponibilizada pelo sítio do Grupo de Estudos em Redes de Computadores (GERCOM)⁹ da Universidade Federal do Pará. Já na camada de Enlace, foi utilizada a implementação padrão do Castalia, chamada *BypassMAC*.

Com relação ao módulo de camada Física, foram incluídos os dados de sensores de temperatura do Laboratório *Intel Berkeley Research (Dataset #5.1)*, do dia 06 a 09 de Março de 2004, totalizando em média 8.640 leituras para cada um dos nós sensores. Embora este *dataset* conte com dados de 54 sensores, apenas 51 foram utilizados. Os sensores 5, 8 e 13, contém erros, como ausência de grande número de detecções, e por isso não foram considerados nas simulações.

A escolha do *Dataset #5.1* se deu pelo fato de conter dados de múltiplos sensores. Dessa forma, é possível reproduzir uma RSSF nas simulações, utilizando

⁹ <http://gercom.ufpa.br>

dados reais, uma vez que a abordagem proposta é baseada nos dados coletados pelos sensores. Desta forma é possível avaliar o potencial da abordagem com relação às métricas redução das transmissões de dados, consumo de energia e atraso.

Nas simulações, foi definido um intervalo de 2 segundos entre o envio de cada detecção através do parâmetro *minSampleInterval*. Dessa forma, para que todos os dados sejam enviados (8.640 detecções), o tempo definido para cada simulação foi de 17.280 segundos. Foram executadas 30 simulações para cada experimento. Na Tabela 5.4 são listados os parâmetros utilizados nas simulações.

Tabela 5.4 – Parâmetros de Simulação

Parâmetro	Valor
Área	150m x 150m
Rádio	CC2420 Texas Instrument
Taxa de Transmissão	250Kbps
Quantidade de sensores	52
Quantidade de sorvedouros	1
Distribuição dos sensores	Aleatória uniforme
Posição do sorvedouro	Central
Padrão de comunicação	IEEE 802.15.4
Raio de alcance	50m
Mobilidade	Não
Intervalo entre detecções	2 segundos
Tempo de simulação	17280 segundos
Energia inicial dos nós	600 Joules
Protocolo de roteamento	Leach
Quantidade de clusters	5

Na Tabela 7.1 são apresentados os resultados obtidos em cada um dos 51 sensores utilizados nas simulações com relação às seguintes métricas: EC: Energia Consumida (*Joules*); PT: Quantidade de Pacotes Transmitidos e MAE: Erro Absoluto Médio (EAM) dos Dados Finais. A Tabela 7.1, encontra-se no Apêndice A. Já no Apêndice B são apresentados diagramas de caixa que demonstram a variação dos resultados observados na Tabela 7.1 com relação à redução de dados e energia consumida pelos nós.

Nas Figuras 5.6 e 5.7, gráficos de barras são apresentados, indicando em

porcentagem os valores obtidos com relação à quantidade de transmissões de dados e consumo médio de energia dos nós nas aplicações *ValueReporting (General)*, abordagem proposta, WMRP, AMDR e QBDR.

Figura 5.6 – Percentual Médio de Transmissões de Dados. Fonte: Próprio Autor

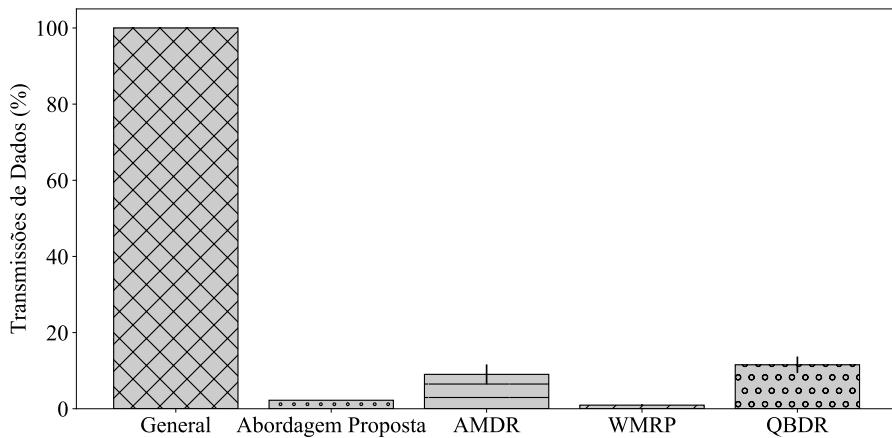
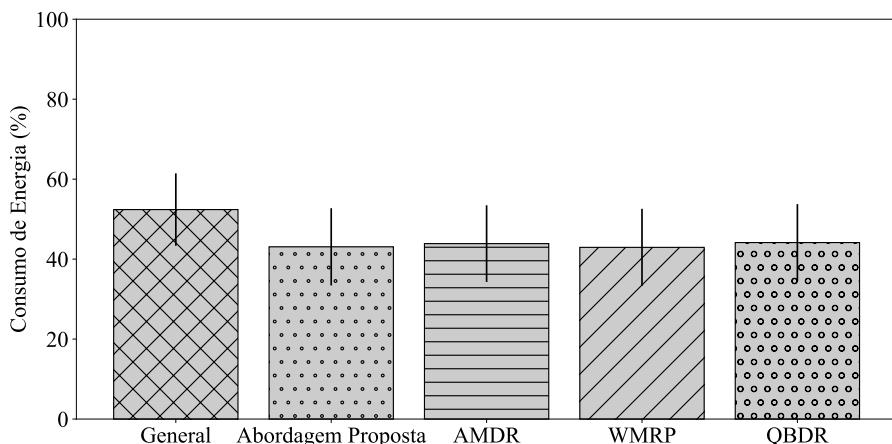


Figura 5.7 – Percentual Médio de Consumo de Energia. Fonte: Próprio Autor



Os resultados apresentados nas Figuras 5.6 e 5.7 mostram que, mesmo ao reduzir significativamente a quantidade de transmissões, o consumo de energia não é reduzido na mesma proporção. Este fato se dá por conta das transições entre modos de operação (RX, TX e *Idle*) realizadas pelo nó sensor. Ou seja, quando a quantidade de transmissões de dados é reduzida, os sensores tendem a permanecer por mais tempo no modo de recepção (RX), o que pode ocasionar um aumento considerável no consumo de energia uma vez que, no rádio CC2420 utilizado no Castalia, o consumo do rádio em modo RX é maior que em modo TX. Nesse sentido, a economia da

energia dos nós é mais significativa quando a quantidade de transmissões é reduzida e o nó sensor permanece por mais tempo em modo *Idle*. As transições entre os modos RX, TX e *Idle* são gerenciados pelo módulo de Enlace do Castalia (BypassMAC) (RASTEGARNIA; SOLOUK, 2011).

Ainda assim, os resultados apresentados na Figura 5.6 e 5.7 mostram que a abordagem proposta, bem como as demais técnicas avaliadas são capazes de reduzir consideravelmente a quantidade de transmissões de dados, proporcionando um uso mais eficiente dos recursos da rede, como largura de banda e energia dos nós.

A média de consumo de energia dos nós sensores em cada método de redução (Abordagem proposta, AMDR, WRMP e QBDR) foi de 258,44J, 263,25J, 257,64J e 264,82J respectivamente. Os testes estatísticos indicam que há significância estatística nestes resultados, sendo que o método WRMP obteve maior desempenho.

Um dos fatores que colaboram para que a técnica *Water Meter Readings Prediction* (WMRP) atinja resultados superiores à abordagem proposta é o comportamento das séries temporais do *dataset* de temperatura (#5.1) do (Laboratório *Intel Berkeley Research*) com relação à estacionariedade. Tais séries possuem longos períodos estacionários e o modelo ARIMA, utilizado na técnica WMRP, alcança bons resultados nestes casos.

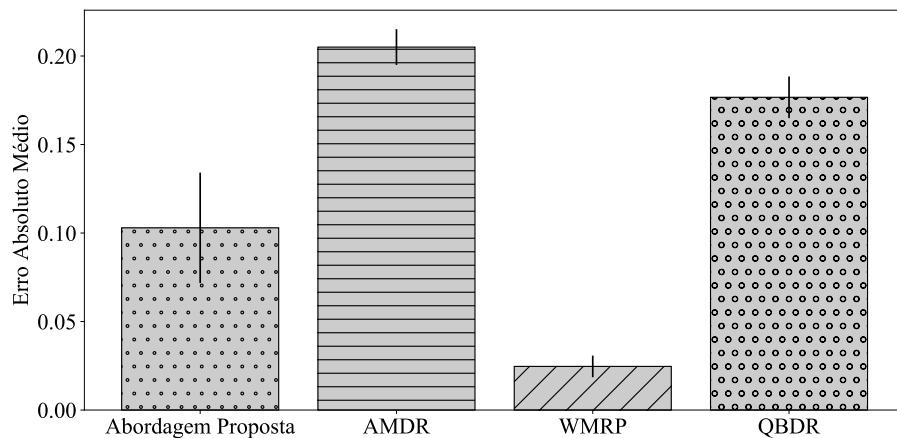
Ainda com relação à estacionariedade das séries temporais, a abordagem proposta conta com um fator limitante que é o parâmetro N_{max} . Este parâmetro é utilizado como um limiar de redução, de forma que o atraso introduzido pela abordagem proposta não seja maior que o máximo requerido pela aplicação de monitoramento.

Já as técnicas *Adaptive Method for Data Reduction* (AMDR) e *Quality-Based Data Reduction* (QBDR), atingiram resultados inferiores com relação à redução de dados na análise do *dataset* #5.1. Ainda assim, o percentual de redução alcançado pelas mesmas é considerável (90,5% e 88,4% respectivamente). Embora estes resultados sejam muito próximos, os testes estatísticos realizados indicam que há significância estatística no resultado obtido pelo AMDR (90,5), sendo estatisticamente superior ao resultado obtido pelo QBDR (88,4).

Na avaliação da acurácia dos dados finais, a aplicação *ValueReporting* não é utilizada, uma vez que não utiliza técnicas para redução de dados, e consequentemente, não apresenta erros nos dados finais. Na Figura 5.8 são apresentados os resultados obtidos com relação ao Erro Absoluto Médio (EAM) dos dados finais, gerados a partir do algoritmo de reconstrução.

Baseado nos resultados apresentados na Figura 5.8, a média de Erro Absoluto Médio (EAM) obtido com o uso da abordagem proposta e dos métodos AMDR, WMRP e QBDR foram 0,10, 0,20, 0,02 e 0,17 respectivamente. Os testes estatísticos indicam

Figura 5.8 – Erro Absoluto Médio (EAM). Fonte: Próprio Autor



que há significância estatística entre os valores de Erro Absoluto Médio (EAM) obtidos em cada uma das técnicas. Nesse sentido, o método *Water Meter Readings Prediction* (WMRP) também obteve os melhores resultados, seguido da abordagem proposta e dos métodos AMDR e QBDR.

Em decorrência das características das RSSF que contém dispositivos de baixa capacidade, alguns fatores requerem análise específica, como: (i) o atraso fim-a-fim; (ii) o atraso introduzido pelo algoritmo de redução; (iii) o tempo de processamento medido na reconstrução, tanto no sensor (fase de inicialização), quanto no sorvedouro.

O atraso fim-a-fim é definido como o tempo medido entre o envio de uma mensagem e a recepção da mesma no nó sorvedouro. Nas simulações, foram medidos os atrasos na entrega das mensagens de todos os nós sensores. Os resultados com relação ao atraso fim-a-fim são apresentados na Figura 5.9.

Identificou-se que o atraso na entrega das mensagens é de 1,43 milissegundos, em média. O desvio padrão é zero, o que significa que, em todas as mensagens enviadas pelos sensores, o tempo até a entrega das mesmas ao sorvedouro é de exatamente 1,43 milissegundos. Justifica-se este resultado dadas as características da rede com relação ao número de nós sensores (51) e tamanho da área (150m x 150m). Como a rede é estática e foi executada com a mesma configuração para todos os métodos avaliados, o atraso obtido foi o mesmo em todos os cenários analisados.

Além do atraso na entrega das mensagens, a abordagem proposta lida com o atraso introduzido pelo algoritmo de redução. Quando o valor de duas ou mais detecções consecutivas são pouco dispersos, elas não são enviadas, mas armazenadas temporariamente no nó sensor, respeitando-se os limiares D_{max} e N_{max} . Dessa forma, a cada transmissão evitada, aumenta-se o tempo entre a realização da detecção e o

Figura 5.9 – Atraso na Entrega das Mensagens. Fonte: Próprio Autor

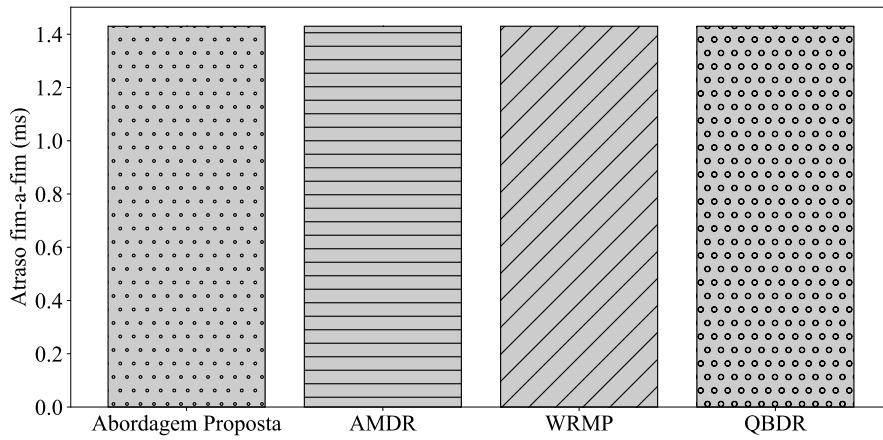
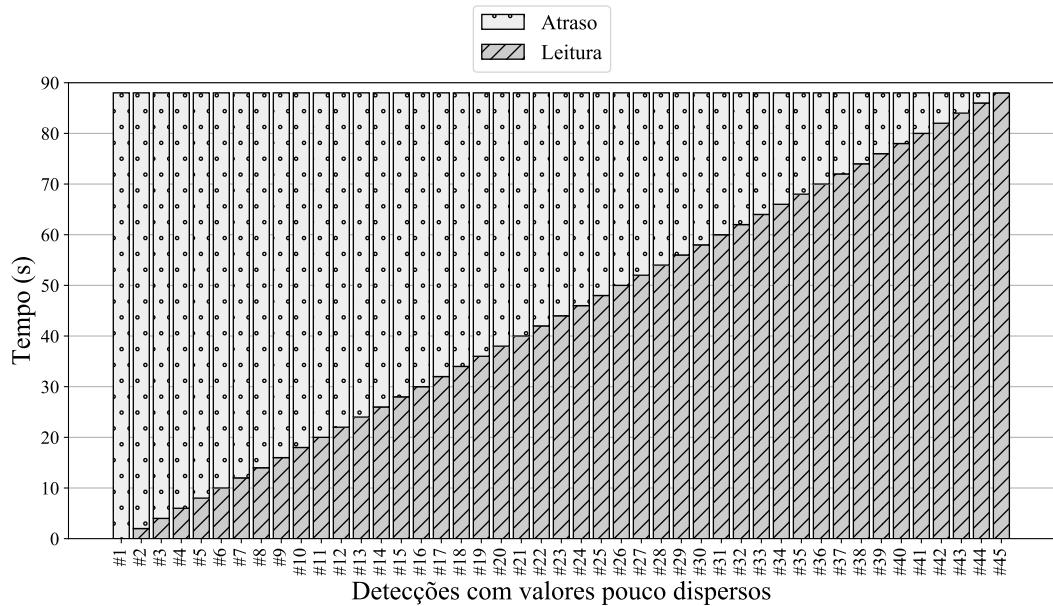


Figura 5.10 – Atraso na Entrega das Mensagens Otimizadas. Fonte: Próprio Autor



seu recebimento pelo nó sorvedouro.

Na Figura 5.10, ilustra-se o atraso introduzido pela abordagem proposta no envio de 1 (uma) mensagem que representa 1 (um) conjunto de detecções com baixa dispersão contendo 45 detecções consecutivas ($N_{max} = 45$) realizadas pelo sensor de número 1 do *dataset* #5.1.

Na Figura 5.10, considera-se que uma nova detecção é realizada a cada dois segundos (eixo x). As barras com hachuras em formato de traços representam o instante de tempo em que a detecção foi realizada (eixo y). Já as barras com hachuras

em formato de pontos representam o tempo medido entre a detecção e a entrega da mesma no nó sorvedouro.

Por exemplo, na Figura 5.10, o tempo entre a realização da detecção #1 e da detecção #45 é de 88 segundos. Isto significa que, embora a frequência dos envios para o sorvedouro seja de dois segundos, o sensor pode ficar sem enviar dados por até 88 segundos, uma vez que os dados coletados neste período foram pouco dispersos.

Conforme os resultados apresentados na Figura 5.10, o atraso vai sendo acumulado em cada detecção, e a média de atraso no conjunto de detecções pouco dispersas pode chegar a 44 segundos, com um desvio padrão de 25,9. Caso o parâmetro N_{max} seja ajustado a um valor muito alto, o percentual de redução a ser alcançado pode ser maior. Contudo o atraso é introduzido de forma linear. Observa-se que o acúmulo do valor do atraso pode implicar em dificuldades para detecção de falhas em um determinado nó sensor.

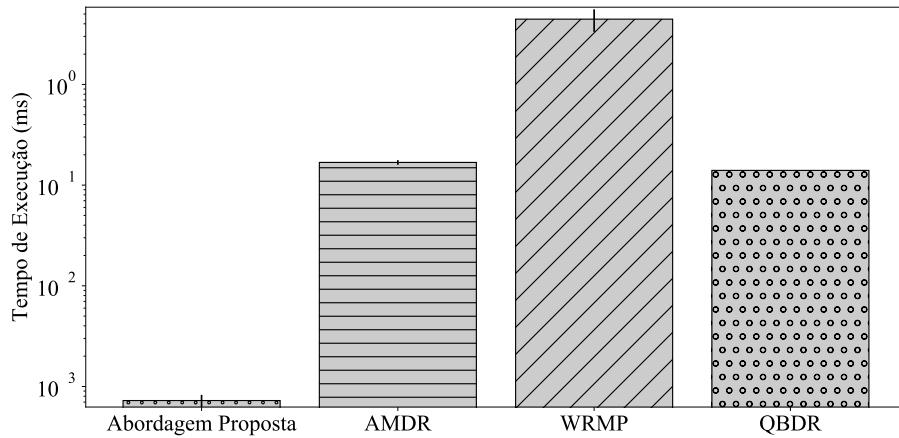
Além disso, dependendo dos requisitos de cada aplicação, o valor de N_{max} deve ser baixo, a fim de garantir que o atraso introduzido pela abordagem proposta não exceda os requisitos da aplicação, como é o caso das aplicações de monitoramento com restrições relacionadas ao tempo. Nesses casos, recomenda-se que o tempo de armazenamento obedeça as restrições de tempo da aplicação.

Outra questão específica a ser analisada é o tempo de processamento da abordagem proposta (Algoritmo 4.1), bem como dos demais métodos de redução de dados (AMDR, WMRP e QBDR). Dessa forma, durante as simulações com o Castalia, foi calculado o tempo de processamento de cada método de redução de dados a cada nova detecção realizada pelo sensor. Ou seja, a cada 2 segundos. A média e desvio padrão do tempo de execução da abordagem proposta e dos demais métodos são apresentados na Figura 5.11.

Os resultados apresentados na Figura 5.11 mostram que a abordagem proposta (Algoritmo 4.1) levou o menor tempo de processamento a cada detecção realizada pelo nó sensor, aproximadamente 0,000725 milissegundos (ms). Já os métodos AMDR, WMRP e QBDR atingiram em média 0,168ms, 4,45ms e 0,140ms respectivamente.

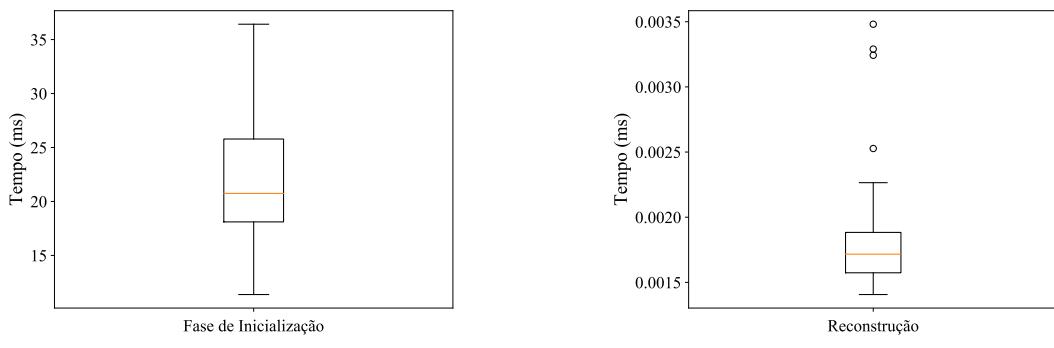
Destaca-se, que a abordagem proposta alcança um alto percentual de redução, com baixo tempo de processamento. Manter um baixo tempo de processamento em sensores é fundamental para estender o tempo de vida da rede, uma vez que o processamento dos dados tem um impacto expressivo no consumo de energia dos dispositivos. Os testes estatísticos demonstram que os resultados não seguem uma distribuição normal, sendo que há significância estatística nos resultados obtidos pela abordagem proposta.

Figura 5.11 – Tempo de Execução. Fonte: Próprio Autor



Além do cálculo do tempo de execução do Algoritmo 4.1, também foi avaliado o tempo de execução do algoritmo de reconstrução de dados que opera no nó sorvedouro (Algoritmo 4.2) e do algoritmo da fase de inicialização para escolha do valor de D_{max} (Algoritmo 4.3), que contém versões modificadas de ambos algoritmos 4.1 e 4.2. Os resultados são apresentados nas Figuras 5.12(a) e 5.12(b).

Figura 5.12 – Tempo de Execução. Fonte: Próprio Autor



(a) Tempo de Execução da Fase de Inicialização

(b) Tempo de Execução da Reconstrução de Dados

Nas simulações, o tempo médio de execução do algoritmo da fase de inicialização (Algoritmo 3.3) foi de 22,27 milissegundos, como demonstra a Figura 5.12(a). É importante destacar, que a fase de inicialização é executada apenas 1(uma) vez quando a rede é ativada para escolha do valor de D_{max} . No decorrer das simulações, após a fase de inicialização, o tempo médio das reconstruções realizadas no nó sorvedouro pelo Algoritmo 4.3, foi de 0,0018ms, como demonstra a Figura 5.12(b).

Pode-se verificar através dos resultados expressos na Figura 5.12, que o tempo de processamento do algoritmo de reconstrução é consideravelmente baixo em relação à frequência em que as detecções são realizadas pelos nós sensores (2 segundos). Dessa forma, pode-se afirmar que o processo de reconstrução não afeta a recepção dos dados pelo nó sorvedouro, nem necessita que o mesmo possua alto poder de processamento.

Segundo a análise dos resultados, verifica-se que a abordagem para a redução baseada na análise de dispersão de dados, superou os demais métodos avaliados em relação ao tempo de processamento, com alto percentual de redução e um baixo erro Erro Absoluto Médio. Além disso, a abordagem levou em média apenas 0,0018ms para realizar a reconstrução dos dados. Este resultado, confirma que a abordagem proposta é uma alternativa tecnológica eficiente e de baixo impacto em relação aos recursos como memória, processamento e energia dos nós sensores.

5.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram apresentados os experimentos e resultados obtidos com a abordagem proposta neste trabalho, em comparação com outros 3 (três) trabalhos encontrados na literatura. São eles: *Adaptive Method for Data Reduction* (AMDR), *Water Meter Readings Prediction* (WMRP) e *Quality-Based Data Reduction* (QBDR).

Os experimentos contaram com 8 conjuntos de dados (*datasets*) reais e publicamente disponíveis a fim de avaliar o desempenho da abordagem proposta com maior eficácia além de realizar uma comparação justa com os trabalhos acima mencionados.

Os resultados obtidos mostram que, na maioria dos *datasets*, a abordagem proposta obteve melhores resultados, representando uma média de 77% de redução de dados. Já os trabalhos AMDR, WMRP e QBDR obtiveram em média a redução de 45,97%, 66,99% e 21,34% dos dados, respectivamente.

Com a finalidade de avaliar a acurácia dos dados reconstruídos pela abordagem proposta com relação à previsão de valores futuros através dos modelos *Recurrent Neural Network* (RNN), *Auto-Regressive Integrated Moving Average* (ARIMA) e *Support Vector Machine* (SVM). Verificou-se que a reconstrução dos dados pode impactar na previsão de valores futuros, visto que foi constatada a diferença estatística entre os resultados de Erro Absoluto Médio (EAM) obtidos entre os dados reais e reconstruídos. Ainda assim, tais resultados não apresentam alta variabilidade, representando em um baixo impacto na predição.

Por fim, foram realizados experimentos com o simulador Castalia, utilizando com base o *dataset* de temperatura do Laboratório *Intel Berkeley Research #5.1*, o

qual possui dados de 54 sensores. Os resultados obtidos mostram que a abordagem proposta e as demais técnicas (AMDR, WMRP e QBDR) obtiveram o consumo médio de 258,44J, 263,25J, 257,64J e 264,82J respectivamente. Dessa forma, a técnica WMRP obteve a menor média de consumo de energia, com significância estatística confirmada pelos testes estatísticos utilizados em relação ao *dataset* #5.1.

Foram avaliados também o tempo de processamento da abordagem proposta, bem como dos demais métodos utilizados. Foi identificado que o algoritmo de redução de dados da abordagem proposta possui o menor tempo de execução, em média 0,000725ms. Já os métodos AMDR, WMRP e QBDR atingiram em média 0,168ms, 4,45ms e 0,140ms respectivamente. Dessa forma, conclui-se que a abordagem proposta pode ser utilizadas em dispositivos com baixo poder de processamento sem exigir alto grau de computação dos mesmos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho de dissertação, foi proposta uma abordagem para a redução da transmissão de dados em nós sensores atuando em aplicações de monitoramento, tais como Redes de Sensores sem Fio (RSSF). O objetivo principal dessa abordagem é evitar que nós sensores transmitam detecções cujos valores demonstram baixa dispersão (variabilidade/espalhamento) dentro de um limite máximo aceitável. Isto significa que os nós sensores apenas transmitirão dados quando o houverem mudanças significativas no ambiente monitorado.

Para alcançar este objetivo, dois algoritmos foram propostos. São eles: Algoritmo para Redução de Dados Baseado na Análise da Dispersão e Algoritmo para Reconstrução de Dados baseado na Tendência Assintótica. O Algoritmo para Redução de Dados Baseado na Análise da Dispersão contém um parâmetro denominado D_{max} que representa a dispersão máxima aceitável para que as transmissões de dados sejam evitadas. Dessa forma, quanto maior o valor de D_{max} , maior redução é alcançada, porém, a acurácia da reconstrução dos dados é mais comprometida. Consequentemente, existe um compromisso entre quantidade de redução e nível de acurácia na reconstrução a ser avaliado.

Uma vez que a abordagem proposta deixa de enviar detecções com baixa dispersão, a frequência em que as transmissões de dados são realizadas pelos sensores é alterada. Dessa forma, são introduzidos atrasos que podem dificultar a identificação de falhas nos nós sensores. Por isso, foi definido um parâmetro chamado N_{max} que estabelece o número de vezes que um nó sensor pode evitar a transmissão de dados, mesmo que os dados coletados sigam apresentando baixa dispersão.

A fim de encontrar o valor ideal para D_{max} , foi proposto um algoritmo que analisa o conjunto das primeiras detecções do nó sensor. Através da análise deste conjunto é possível identificar o nível de dispersão dos dados coletados, e com base nisso, definir o valor ideal para D_{max} levando em consideração um valor de erro máximo aceitável e_{max} .

De acordo com as características das aplicações de monitoramento, os valores definidos para os parâmetros N_{max} e e_{max} podem mudar. Dessa forma, foi proposta uma taxonomia que atribui valores específicos para N_{max} e e_{max} de acordo com os requisitos de cada aplicação de monitoramento, como atraso na entrega das mensagens e erro máximo aceitável.

Nos experimentos foram utilizadas 23 séries temporais provenientes de 8 conjuntos de dados (*datasets*) reais de aplicações de monitoramento. Para fins de com-

paração com a abordagem proposta, foram implementados os trabalhos *Adaptive Method for Data Reduction* (AMDR) (FATHY; BARNAGHI; TAFAZOLLI, 2018), *Water Meter Readings Prediction* (WMRP) (ALMHANA; CHOULAKIAN; ALMHANA, 2017) e *Quality-Based Data Reduction* (QBDR) (SANTINI; ROMER, 2006).

Os experimentos foram divididos em três etapas. Na primeira etapa, foi analisado o potencial da abordagem proposta, bem como dos métodos AMDR, WMRP e QBDR. Os experimentos foram conduzidos de forma que os *datasets* compõe a entrada dos algoritmos de redução, que por sua vez, têm a finalidade processar os dados e verificar o percentual de mensagens que poderiam ser evitadas, bem como, avaliar a acurácia dos dados gerados pelo algoritmo de reconstrução proposto. Os resultados obtidos demonstram que a abordagem proposta obteve os melhores resultados na maioria dos *datasets*, sendo que a significância estatística foi atestada pelos testes estatísticos.

Na segunda etapa, foi avaliada a acurácia dos dados reconstruídos através da análise dos modelos de predição Rede Neural Recorrente (RNN), *Auto-Regressive Integrated Moving Average* (ARIMA) e *Support Vector Machine* (SVM). Os resultados indicam que o modelo ARIMA alcançou predições mais assertivas no âmbito geral, se comparado com os modelos Rede Neural Recorrente e SVM.

Por fim, foram realizadas simulações no Simulador Castalia, a fim de verificar o percentual de detecções suprimidas, bem como o consumo energético dos nós sensores e o atraso introduzido pela abordagem proposta no contexto de uma Rede de Sensores sem Fio.

No Castalia, foram incluídos os dados dos 54 sensores de temperatura do Laboratório *Intel Berkeley Research Lab* (#5.1). Os resultados obtidos mostram que o método *Water Meter Readings Prediction* (WMRP) atingiu maior nível de redução de dados, e consequentemente, maior economia de energia dos nós, se comparado à abordagem proposta e aos demais métodos. Este resultado dá-se por conta do comportamento da série temporal do *dataset* #5.1, que é caracterizado pela baixa dispersão e estacionariedade. Contudo, a abordagem proposta tende a apresentar melhores resultados com séries temporais sazonais.

Além disso, o tempo de processamento da abordagem proposta e dos métodos AMDR, WMRP e QBDR foram avaliados. A cada detecção realizada pelos nós sensores, os métodos de redução de dados são executados. O algoritmo de redução de dados da abordagem proposta obteve o menor tempo de execução, se comparado aos demais métodos. Além disso, foi avaliado também o tempo médio de execução do algoritmo de reconstrução da abordagem proposta, que foi em média 0,0018 milissegundos. Os demais métodos de redução de dados avaliados não oferecem a etapa de

reconstrução. Dessa forma, conclui-se que o processo de reconstrução da abordagem proposta não afeta a recepção dos dados pelo nó sorvedouro, podendo ser facilmente aplicado em dispositivos com baixo poder de processamento.

6.1 TRABALHOS FUTUROS

Como perspectivas de trabalhos futuros, sugere-se a implementação de uma abordagem híbrida para redução de dados em Redes de Sensores sem Fio (RSSF), baseando-se tanto na análise da dispersão, quanto na predição de valores futuros, a fim de alcançar maior nível de redução nas transmissões, e consequentemente, maior economia de energia dos nós.

Além disto, sugere-se utilizar outros *datasets* que contenham grande quantidade de dados e cujas séries temporais demonstrem diferentes comportamentos, como tendência, aleatoriedade e sazonalidade.

É relevante também avaliar o consumo energético dos nós sensores identificando o percentual de energia gasto com os modos de transmissão (TX), recepção (RX) e Idle.

6.2 OUTRAS CONTRIBUIÇÕES DA DISSERTAÇÃO

Os resultados obtidos durante a pesquisa realizadas no mestrado foram publicadas nas seguintes conferências:

- S. Oliveira, J. Kniess, R. Parpinelli e W. Castañeda, “Predição de Séries Temporais em Internet das Coisas com Redes Neurais Recorrentes”, In: **50º Simpósio Brasileiro de Pesquisa Operacional (SBPO)**, 2018. (OLIVEIRA et al., 2018).
- S. Oliveira, J. Kniess, V. Marques. “A Method Based on Dispersion Analysis for Data Reduction in WSN”, In: **Proceedings of International Conference on Green, Pervasive, and Cloud Computing (GPC)**, 2019. (OLIVEIRA; KNIESS; MARQUES, 2019).
- S. Oliveira, J. Kniess. “Abordagem Baseada em Análise de Dispersão para Redução de Dados em Redes de Sensores sem Fio”, In: **Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**, 2019. Gramado, RS. (OLIVEIRA; KNIESS, 2019)

7 APÊNDICE A - RESULTADOS OBTIDOS NO CASTALIA

Neste Apêndice apresenta-se a tabela com os resultados obtidos no Castalia, referentes à energia consumida (EC), à quantidade de pacotes transmitidos (PT) e Erro Absoluto Médio (EAM) dos dados finais.

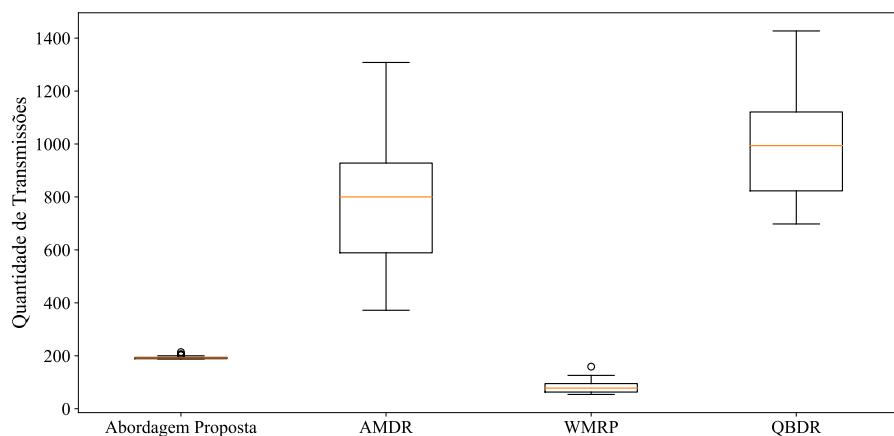
Tabela 7.1 – Resultados obtidos no Castalia

	ValueReporting		Abordagem Proposta			WMRP			AMDR			QBDR		
	EC	PT	EC	PT	MAE	EC	PT	MAE	EC	PT	MAE	EC	PT	MAE
node_1	299,582	8640	239,555	188	0,09	237,263	62	0,02562	299,582	624	0,20137	243,973	834	0,17108
node_2	475,844	8640	431,288	190	0,077	430,909	61	0,01986	475,844	556	0,21408	433,91	781	0,17907
node_3	292,092	8640	236,081	189	0,068	235,581	57	0,0235	292,092	543	0,21288	241,654	825	0,19813
node_4	309,934	8640	258,464	193	0,062	257,74	60	0,0197	309,934	525	0,21741	261,554	795	0,17495
node_6	376,765	8640	328,661	189	0,095	325,925	61	0,0207	376,765	588	0,20926	332,349	878	0,1848
node_7	291,705	8640	228,487	188	0,0587	228,17	54	0,01382	291,705	372	0,22662	234,724	698	0,18468
node_9	268,253	8641	207,958	189	0,08	207,626	63	0,01859	268,253	545	0,22032	214,293	787	0,19783
node_10	265,491	8640	213,094	189	0,073	212,696	63	0,0189	265,491	451	0,19996	215,719	749	0,18603
node_11	327,556	8640	278,825	190	0,102	277,928	71	0,02601	327,556	624	0,2001	283,084	948	0,16571
node_12	312,962	8640	253,471	206	0,114	253,134	86	0,02738	312,962	945	0,19132	263,803	1206	0,17256
node_13	278,123	8641	218,294	194	0,108	218,055	89	0,03331	278,123	825	0,18945	227,037	1149	0,17741
node_14	293,224	8639	244,726	196	0,097	241,694	75	0,0261	293,224	819	0,20256	248,299	994	0,16984
node_16	342,555	8639	292,255	200	0,14	292,121	103	0,0379	342,555	881	0,1865	301,013	1304	0,16764
node_17	302,5	8640	243,446	193	0,095	241,961	92	0,02453	302,5	720	0,198	248,837	1048	0,16962
node_18	534,088	8639	492,15	188	0,0484	490,618	57	0,01429	534,088	477	0,22456	496,842	736	0,20858
node_19	268,895	8641	205,657	194	0,1	203,672	68	0,02591	268,895	775	0,20441	210,703	871	0,1492
node_20	281,092	8640	219,989	193	0,081	219,639	68	0,01774	281,092	542	0,2164	228,345	1007	0,17847
node_21	336,706	8640	285,356	191	0,104	283,966	69	0,02852	336,706	779	0,20686	292,9	1107	0,19669
node_22	278,841	8640	218,691	189	0,081	218,428	64	0,0194	278,841	625	0,2147	224,802	891	0,17494
node_23	290,043	8640	240,849	188	0,072	239,419	65	0,0194	290,043	590	0,21972	245,075	769	0,1882
node_24	380,823	8640	327,794	203	0,107	327,174	120	0,0249	380,823	927	0,20682	333,423	1062	0,17135
node_25	275,327	8640	211,869	203	0,116	211,776	159	0,0239	275,327	964	0,19794	218,108	1142	0,18556
node_26	261,934	8641	205,403	191	0,101	204,514	126	0,0254	261,934	929	0,2044	210,335	1028	0,17147
node_27	278,29	8640	219,022	196	0,144	218,845	97	0,03105	278,29	1294	0,19088	226,437	1266	0,16789
node_28	272,753	8640	211,753	197	0,091	210,003	88	0,0238	272,753	800	0,211	217,024	993	0,17874
node_29	320,845	8640	266,878	191	0,102	264,063	93	0,0225	320,845	827	0,21095	273,039	946	0,16614
node_30	307,437	8640	256,267	191	0,147	256,086	94	0,034	307,437	1292	0,19825	265,525	1321	0,17208
node_31	298,74	8640	246,947	189	0,073	245,126	60	0,0182	298,74	622	0,21489	250,4	798	0,19542
node_32	315,376	8640	258,036	193	0,15	255,96	100	0,03532	315,376	1207	0,18961	264,899	1310	0,14803
node_33	285,082	8640	219,63	192	0,143	219,542	119	0,0423	285,082	1308	0,18996	229,336	1394	0,17419
node_34	330,217	8641	260,736	192	0,133	260,498	95	0,02805	330,217	914	0,1972	266,958	1096	0,17456
node_35	305,341	8640	245,833	194	0,068	245,346	68	0,01931	305,341	532	0,21065	252,166	809	0,20354
node_36	298,254	8640	238,092	205	0,08	237,783	80	0,02174	298,254	601	0,2084	243,757	792	0,18679
node_37	285,76	8640	235,982	188	0,069	235,762	61	0,02063	285,76	542	0,21584	241,436	806	0,16992
node_38	328,675	8640	276,427	194	0,096	276,274	98	0,02914	328,675	787	0,19845	282,562	1006	0,17114
node_39	464,624	8639	414,213	192	0,123	413,889	90	0,0273	464,624	877	0,20335	422,352	971	0,1815
node_40	313,663	8640	258,546	188	0,116	258,099	97	0,02767	313,663	954	0,2052	266,351	1128	0,1746
node_41	305,748	8640	244,594	193	0,124	244,351	95	0,02723	305,748	981	0,20317	253,418	1104	0,17973
node_42	293,306	8640	241,476	194	0,14	241,278	90	0,0295	293,306	977	0,20306	249,174	1154	0,18093
node_43	280,771	8640	220,799	189	0,113	220,638	76	0,025	280,771	852	0,20316	228,969	1030	0,18135
node_44	289,045	8640	224,317	214	0,099	223,955	122	0,02275	289,045	860	0,2032	230,584	988	0,16748
node_45	285,86	8640	234,805	189	0,062	233,152	59	0,01543	285,86	570	0,21898	238,632	821	0,17993
node_46	278,653	8641	210,711	191	0,068	210,375	63	0,01844	278,653	605	0,21507	216,231	869	0,17996
node_47	366,153	8640	307,497	189	0,09	307,143	59	0,0212	366,153	683	0,21644	314,799	888	0,17544
node_48	288,859	8640	233,72	189	0,057	233,471	55	0,01452	288,859	522	0,21335	239,356	706	0,17508
node_49	291,912	8640	237,64	194	0,142	237,498	106	0,03015	291,912	977	0,19723	246,15	1114	0,16549
node_50	300,972	8640	240,94	196	0,195	238,665	104	0,0354	300,972	1156	0,19106	254,378	1427	0,16692
node_51	362,041	8639	313,908	192	0,146	313,463	87	0,02666	362,041	835	0,19605	320,6	1100	0,16132
node_52	396,036	8639	344,668	190	0,145	344,46	80	0,027	396,036	849	0,18793	353,259	1144	0,17253
node_53	261,613	8641	204,188	192	0,147	203,857	78	0,0275	261,613	937	0,19465	213,154	1228	0,16526
node_54	281,111	8640	230,545	189	0,116	230,167	69	0,0199	281,111	807	0,20529	234,57	1052	0,17085

8 APÊNDICE B - CONSIDERAÇÕES SOBRE A REDUÇÃO DE DADOS E CONSUMO DE ENERGIA NO CASTALIA

Nas Figuras 8.1 e 8.2 são apresentados, através de diagramas de caixa, os resultados obtidos com relação à quantidade de transmissões, consumo de energia e Erro Absoluto Médio (EAM), a fim de possibilitar melhor visualização e análise dos resultados descritos na Tabela 7.1.

Figura 8.1 – Quantidade de Transmissões. Fonte: Próprio Autor

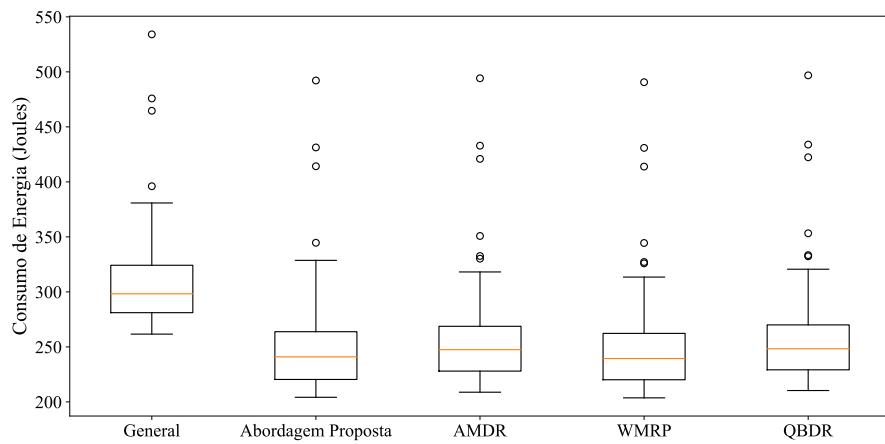


Na Figura 8.1, a aplicação *ValueReporting* não é apresentada pelo fato de realizar 100% das transmissões de dados, o que representa a quantidade de 8.640 detecções para cada sensor.

Os resultados ilustrados na Figura 8.1 mostram que a abordagem proposta e a técnica *Water Meter Readings Prediction* (WMRP) alcançam maior nível de redução (97,6% e 98,84% respectivamente), proporcionando um uso mais eficiente dos recursos da rede. De acordo com os testes estatísticos realizados, foi atestada a diferença estatística entre os resultados, sendo o método WMRP o que apresenta os melhores resultados (98,84%).

Já na Figura 8.2, pode-se verificar que a aplicação padrão do Castalia (*General*) consome mais energia, visto que nenhuma técnica de redução de dados é empregada. Além disso, o percentual de energia consumida pelos nós sensores através da abordagem proposta e das demais técnicas não varia consideravelmente. Ainda assim, os testes estatísticos atestam diferença estatística entre todas as técnicas. Como consequência do alto desempenho na redução das transmissões de dados, a técnica WMRP também obteve os melhores resultados com relação ao consumo energético.

Figura 8.2 – Consumo de Energia dos Nós Sensores. Fonte: Próprio Autor



Outra questão a ser analisada na Figura 8.2 é a existência de valores atípicos. Em todos os casos, foram observados valores atípicos (discrepantes), representados com pontos individuais. Isto significa que houve uma pequena quantidade de sensores que consumiram um percentual maior de energia se comparado aos demais.

REFERÊNCIAS

- A hierarchical fractional LMS prediction method for data reduction in a wireless sensor network. *Ad Hoc Networks*, v. 87, p. 113 – 127, 2019.
- ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. **arXiv preprint arXiv:1603.04467**, Ithaca, NY, 2016.
- ABDELGAWAD, A. M. **Resourse-aware Data Fusion Algorithms for Wireless Sensor Networks**. Tese (Doutorado) — University of Louisiana at Lafayette, 2011. AAI3496493.
- ADU-MANU, K. S.; TAPPARELLO, C.; HEINZELMAN, W.; KATSRIKU, F. A.; ABDULAI, J.-D. Water quality monitoring using wireless sensor networks: Current trends and future research directions. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 13, n. 1, p. 4:1–4:41, jan. 2017. ISSN 1550-4859.
- AL-HOQANI, N.; YANG, S.-H. Adaptive sampling for wireless household water consumption monitoring. **Procedia Engineering**, v. 119, p. 1356 – 1365, 2015. Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management.
- ALBUQUERQUE, P. H. **Previsão de séries temporais financeiras por meio de máquinas de suporte vetorial e ondaletas**. Tese (Doutorado) — Universidade de São Paulo, São Paulo, SP, 2014.
- ALI, N. A.; DRIEBERG, M.; SEBASTIAN, P. Deployment of micaz mote for wireless sensor network applications. In: **2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)**. Penang, Malaysia: [s.n.], 2011.
- ALMHANA, C.; CHOULAKIAN, V.; ALMHANA, J. An efficient approach for data transmission in power-constrained wireless sensor network. In: **2017 IEEE International Conference on Communications (ICC)**. Paris, France: [s.n.], 2017. p. 1–6.
- ALSHEIKH, M. A.; LIN, S.; NIYATO, D.; TAN, H.-P. Rate-distortion balanced data compression for wireless sensor networks. **IEEE Sensors Journal**, Institute of Electrical and Electronics Engineers (IEEE), v. 16, n. 12, p. 5072–5083, jun 2016.
- ALSHEIKH, M. A.; LIN, S.; TAN, H. P.; NIYATO, D. Toward a robust sparse data representation for wireless sensor networks. In: **2015 IEEE 40th Conference on Local Computer Networks (LCN)**. Clearwater Beach, FL, USA: [s.n.], 2015. p. 117–124.
- ANASTASI, G.; CONTI, M.; FRANCESCO, M. D.; PASSARELLA, A. Energy conservation in wireless sensor networks: A survey. **Ad Hoc Networks**, v. 7, n. 3, p. 537 – 568, 2009.
- ANDRÉ, L.; PARPINELLI, R. S. Tutorial sobre o uso de técnicas para controle de parâmetros em algoritmos de inteligência de enxame e computação evolutiva. **Revista de Informática Teórica e Aplicada**, v. 21, n. 2, p. 90–135, 2014.

ANG, K. H.; CHONG, G.; LI, Y. Pid control system analysis, design, and technology. **IEEE Transactions on Control Systems Technology**, v. 13, n. 4, p. 559–576, July 2005. ISSN 1063-6536.

ARASU, K.; GANESAN, R. Effective implementation of energy aware routing for wireless sensor network. **Materials Today: Proceedings**, v. 5, n. 1, Part 1, p. 1186 – 1193, 2018. International Conference on Processing of Materials, Minerals and Energy (July 29th – 30th) 2016, Ongole, Andhra Pradesh, India.

ASHTON, K. **That 'Internet of Things' Thing**. 2009. <<http://www.rfidjournal.com/articles/view?4986>>. [Online; acessado em 15/05/2018].

BAL, M.; SHEN, W.; GHENNIWA, H. Collaborative signal and information processing in wireless sensor networks: A review. In: **2009 IEEE International Conference on Systems, Man and Cybernetics**. San Antonio, TX, USA: [s.n.], 2009. p. 3151–3156.

BASAK, D.; PAL, S.; PATRANABIS, D. C. Support vector regression. **Neural Information Processing-Letters and Reviews**, Zurich, Switzerland, v. 11, n. 10, p. 203–224, 2007.

BASTIEN, F.; LAMBLIN, P.; PASCANU, R.; BERGSTRA, J.; GOODFELLOW, I. J.; BERGERON, A.; BOUCHARD, N.; WARDE-FARLEY, D.; BENGIO, Y. Theano: new features and speed improvements. **CoRR**, Ithaca, NY, v. 1211.5590, 2012.

BORGNE, Y.-A. L.; SANTINI, S.; BONTEMPI, G. Adaptive model selection for time series prediction in wireless sensor networks. **Signal Processing**, v. 87, n. 12, p. 3010 – 3020, 2007.

BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: **Proceedings of COMPSTAT'2010**. Heidelberg: Springer, 2010. p. 177–186.

BOUKARY, N. A. **A COMPARISON OF TIME SERIES FORECASTING LEARNING ALGORITHMS ON THE TASK OF PREDICTING EVENT TIMING**. Tese (Doutorado) — Royal Military College of Canada, 2016.

BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time series analysis: forecasting and control**. San Francisco, CA: John Wiley & Sons, 2015.

CALLEGARO, R. F. **Uma arquitetura para fusão de dados de sensores de baixo custo em redes de sensores sem fio**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, 2 2014.

CANDÈS, E. J.; ROMBERG, J.; TAO, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. **IEEE Transactions on information theory**, IEEE, v. 52, n. 2, p. 489–509, 2006.

CANDES, E. J.; WAKIN, M. B. An introduction to compressive sampling. **IEEE Signal Processing Magazine**, v. 25, n. 2, p. 21–30, March 2008. ISSN 1053-5888.

CASTAÑEDA, W. A. C. **Metodología de gestión ubíqua para tecnología médico-hospitalar utilizando tecnologías pervasivas**. Tese (Doutorado) — Federal University of Santa Catarina, Florianópolis, Brazil, 2016.

CHAKRABORTY, I.; DAS, P. Data fusion in wireless sensor network-a survey. **International Journal of Scientific Research in Network Security and Communication**, ISROSET: International Scientific Research Organization for Science, Engineering and Technology, v. 5, n. 6, p. 9–15, dec 2017.

CHEN, J.; WANG, N.; CHEN, D.; WAN, J. An adaptive dynamic topology wsn data collection method based on compressive sensing. In: **2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)**. Xian, China: [s.n.], 2018. p. 1376–1379.

CHEN, Y.; SHEN, C.; ZHANG, K.; WANG, H.; GAO, Q. Leach algorithm based on energy consumption equilibrium. In: **2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)**. Xiamen, China: [s.n.], 2018. p. 677–680.

CITY OF CHICAGO. **Beach Water Quality - Automated Sensors**. 2019. <<https://data.cityofchicago.org/Parks-Recreation/Beach-Water-Quality-Automated-Sensors/qmz-2xku>>. [Online; acessado em 14/02/2019].

CITY OF CHICAGO. **Beach Weather Stations - Automated Sensors**. 2019. <<https://data.cityofchicago.org/Parks-Recreation/Beach-Weather-Stations-Automated-Sensors/k7hf-8y75>>. [Online; acessado em 14/02/2019].

COLLOBERT, R.; FARABET, C.; KAVUKCUOĞLU, K. Torch| scientific computing for luajit. In: **NIPS Workshop on Machine Learning Open Source Software**. Ithaca, NY: [s.n.], 2008. v. 1, n. 1211.5590.

DAFLAPURKAR, P. M.; GANDHI, M.; PATIL, B. Tree based distributed clustering routing scheme for energy efficiency in wireless sensor networks. In: **IEEE. 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)**. Chennai, India, 2017. p. 2450–2456.

DAGAR, M.; MAHAJAN, S. Data aggregation in wireless sensor network: a survey. **International Journal of Information and Computation Technology**, v. 3, n. 3, p. 167–174, 2013.

DEB, C.; ZHANG, F.; YANG, J.; LEE, S. E.; SHAH, K. W. A review on time series forecasting techniques for building energy consumption. **Renewable and Sustainable Energy Reviews**, v. 74, p. 902 – 924, 2017.

DERTAT, A. **Applied Deep Learning - Part 1: Artificial Neural Networks**. 2017. <<https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6>>. [Online; acessado em: 14/09/2018].

DHAND, G.; TYAGI, S. Data aggregation techniques in WSN:survey. **Procedia Computer Science**, Elsevier BV, v. 92, p. 378–384, 2016.

DHASIAN, H. R.; BALASUBRAMANIAN, P. Survey of data aggregation techniques using soft computing in wireless sensor networks. **IET Information Security**, v. 7, n. 4, p. 336–342, December 2013. ISSN 1751-8709.

DIAS, G. M. **Prediction-based strategies for reducing data transmissions in the IoT**. Tese (Doutorado) — Universitat Pompeu Fabra, 2016.

DIAS, G. M.; BELLALTA, B.; OECHSNER, S. A survey about prediction-based data reduction in wireless sensor networks. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 49, n. 3, p. 58:1–58:35, nov. 2016.

DIAS, G. M.; BELLALTA, B.; OECHSNER, S. Using data prediction techniques to reduce data transmissions in the IoT. In: **2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)**. Reston, VA, USA: IEEE, 2016.

DREDZE, M.; CRAMMER, K.; PEREIRA, F. Confidence-weighted linear classification. In: **Proceedings of the 25th International Conference on Machine Learning**. New York, NY, USA: ACM, 2008. (ICML '08), p. 264–271.

DRUCKER, H.; BURGES, C. J.; KAUFMAN, L.; SMOLA, A. J.; VAPNIK, V. Support vector regression machines. In: **Advances in neural information processing systems**. [S.I.: s.n.], 1997. p. 155–161.

EL-TELBANY, M. E.; MAGED, M. A. Exploiting sparsity in wireless sensor networks for energy saving: A comparative study. **International Journal of Applied Engineering Research**, v. 12, n. 4, p. 452–460, 2017.

ELSHRKAWEY, M.; ELSHERIF, S. M.; WAHED, M. E. An enhancement approach for reducing the energy consumption in wireless sensor networks. **Journal of King Saud University - Computer and Information Sciences**, v. 30, n. 2, p. 259 – 267, 2018.

FADAEDDINI, A.; ESHGHI, M.; MAJIDI, B. A deep residual neural network for low altitude remote sensing image classification. In: **2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)**. Kerman, Iran: [s.n.], 2018.

FARHATH, Z. A.; ARPUTHAMARY, B.; AROCKIAM, D. L. **A Survey on ARIMA Forecasting Using Time Series Model**. [S.I.]: International Journal of Computer Science and Mobile Computing, 2016.

FATHY, Y.; BARNAGHI, P.; TAFAZOLLI, R. An adaptive method for data reduction in the internet of things. In: IEEE. **Proceedings of IEEE 4th World Forum on Internet of Things**. Singapore, Singapore, 2018.

FEHRI, C. E.; KASSAB, M.; ABDELLATIF, S.; BERTHOU, P.; BELGHITH, A. Lora technology mac layer operations and research issues. **Procedia Computer Science**, v. 130, p. 1096 – 1101, 2018. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.

GANJEWAR, P. D.; BARANI, S.; WAGH, S. J.; SONAVANE, S. S. Survey on data reduction techniques for energy conservation for prolonging life of wireless sensor network. **Wireless Communication**, v. 10, n. 2, p. 17–25, 2018.

GARTNER. **Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016**. 2017. <<https://www.gartner.com/newsroom/id/3598917>>. [Online; acessado em 05/12/2017].

GERVEN, M. van; BOHTE, S. **Artificial neural networks as models of neural information processing**. [S.I.]: Frontiers Media SA, 2018.

- GHICA, O. C.; TRAJCEVSKI, G.; SCHEUERMANN, P.; BISCHOF, Z.; VALTCHANOV, N. Sidnet-swans: A simulator and integrated development platform for sensor networks applications. In: **Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems**. New York, NY, USA: ACM, 2008. (SenSys '08), p. 385–386. ISBN 978-1-59593-990-6.
- GILBERT, E. P. K.; KALIAPERUMAL, B.; RAJSINGH, E. B.; LYDIA, M. Trust based data prediction, aggregation and reconstruction using compressed sensing for clustered wireless sensor networks. **Computers & Electrical Engineering**, Elsevier BV, feb 2018.
- GOLDBERG, Y. Neural network methods for natural language processing. **Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers LLC, v. 10, n. 1, p. 1–309, apr 2017.
- GOMBÉ, B. O.; MÉROU, G. G.; BRESCHI, K.; GUYENNET, H.; FRIEDT, J.-M.; FELEA, V.; MEDJAHER, K. A saw wireless sensor network platform for industrial predictive maintenance. **Journal of Intelligent Manufacturing**, Jul 2017.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.I.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.
- GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of things (IoT): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, Elsevier BV, v. 29, n. 7, p. 1645–1660, sep 2013.
- GUPTA, G. P.; JHA, S. Integrated clustering and routing protocol for wireless sensor networks using cuckoo and harmony search based metaheuristic techniques. **Engineering Applications of Artificial Intelligence**, v. 68, p. 101 – 109, 2018. ISSN 0952-1976.
- GUPTA, M.; SHUM, L. V.; BODANESE, E.; HAILES, S. Design and evaluation of an adaptive sampling strategy for a wireless air pollution sensor network. In: **2011 IEEE 36th Conference on Local Computer Networks**. [S.I.: s.n.], 2011. p. 1003–1010.
- HA, S.; RHEE, I.; XU, L. Cubic: A new tcp-friendly high-speed tcp variant. **SIGOPS Oper. Syst. Rev.**, ACM, New York, NY, USA, v. 42, n. 5, p. 64–74, jul. 2008. ISSN 0163-5980.
- HE, X.; XU, S. Artificial neural networks. **Process Neural Networks: Theory and Applications**, Springer, p. 20–42, 2010.
- HUANG, Z.; LI, M.; SONG, Y.; ZHANG, Y.; CHEN, Z. Adaptive compressive data gathering for wireless sensor networks. In: **2017 3rd IEEE International Conference on Computer and Communications (ICCC)**. Chengdu, China: [s.n.], 2017. p. 362–367.
- HUGHES, J.; YAN, J.; SOGA, K. DEVELOPMENT OF WIRELESS SENSOR NETWORK USING BLUETOOTH LOW ENERGY (BLE) FOR CONSTRUCTION NOISE MONITORING. **International Journal on Smart Sensing and Intelligent Systems**, Exeley, Inc., v. 8, n. 2, p. 1379–1405, 2015.
- HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: principles and practice**. [S.I.]: OTexts, 2018.

IMDADULLAH, M. **Coefficient of Determination: A model Selection Criteria.** 2012. <<http://itfeature.com/correlation-and-regression-analysis/coefficients-of-determination>>. [Online; acessado em 13/07/2018].

INGELREST, F.; BARRENETXEA, G.; SCHAEFER, G.; VETTERLI, M.; COUACH, O.; PARLANGE, M. Sensorscope: Application-specific sensor network for environmental monitoring. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 6, n. 2, p. 17:1–17:32, mar. 2010. ISSN 1550-4859.

JI, S.; CARIN, L. Bayesian compressive sensing and projection optimization. In: **Proceedings of the 24th International Conference on Machine Learning**. New York, NY, USA: ACM, 2007. (ICML '07), p. 377–384. ISBN 978-1-59593-793-3.

KARIM, S. **Energy Efficiency in Wireless Sensor Networks, Through Data Compression.** Dissertação (Mestrado) — Universitetet i Oslo, Oslo, Norway, 2017.

KHO, J.; ROGERS, A.; JENNINGS, N. R. Decentralized control of adaptive sampling in wireless sensor networks. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 5, n. 3, p. 19:1–19:35, jun. 2009. ISSN 1550-4859.

KRAUSE, E. F. **Taxicab geometry: An adventure in non-Euclidean geometry.** [S.l.]: Courier Corporation, 1986.

KRISHNAMACHARI, L.; ESTRIN, D.; WICKER, S. The impact of data aggregation in wireless sensor networks. In: **Proceedings 22nd International Conference on Distributed Computing Systems Workshops**. Vienna, Austria, Austria: [s.n.], 2002. p. 575–578.

LANG, K.; ZHANG, M.; YUAN, Y.; YUE, X. Short-term load forecasting based on multivariate time series prediction and weighted neural network with random weights and kernels. **Cluster Computing**, Feb 2018.

LEE, I.; LEE, K. The internet of things (iot): Applications, investments, and challenges for enterprises. **Business Horizons**, v. 58, n. 4, p. 431 – 440, 2015.

LI, S.; CUI, J.; LI, Z. Wireless sensor network for precise agriculture monitoring. In: **2011 Fourth International Conference on Intelligent Computation Technology and Automation**. Shenzhen, Guangdong, China: [s.n.], 2011. v. 1, p. 307–310.

LI, Z.; ZHANG, W.; QIAO, D.; PENG, Y. Lifetime balanced data aggregation for the internet of things. **Computers & Electrical Engineering**, Elsevier BV, Beijing, China, v. 58, p. 244–264, feb 2017.

LIU, Y.; QIAN, K. A novel tree-based routing protocol in zigbee wireless networks. In: **2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)**. [S.l.: s.n.], 2016. p. 469–473.

LIU, Z.; SULLIVAN, C. J. Prediction of weather induced background radiation fluctuation with recurrent neural networks. **Radiation Physics and Chemistry**, Elsevier BV, mar 2018.

LIU, Z. ya. Hardware design of smart home system based on zigbee wireless sensor network. **AASRI Procedia**, v. 8, p. 75 – 81, 2014. ISSN 2212-6716. 2014 AASRI Conference on Sports Engineering and Computer Science (SECS 2014).

- LOKESH, S.; KUMAR, P. M.; DEVI, M. R.; PARTHASARATHY, P.; GOKULNATH, C. An automatic tamil speech recognition system by using bidirectional recurrent neural network with self-organizing map. **Neural Computing and Applications**, Apr 2018.
- LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007.
- LUO, J.; XIANG, L.; ROSENBERG, C. Does compressed sensing improve the throughput of wireless sensor networks? In: **2010 IEEE International Conference on Communications**. Cape Town, South Africa: [s.n.], 2010. p. 1–6.
- MADDEN, S. **Intel Lab Data**. 2004. <<http://db.lcs.mit.edu/labdata/labdata.html>>. [Online; acessado em 08/03/2018].
- MAMUN, Q. A qualitative comparison of different logical topologies for wireless sensor networks. v. 12, p. 14887–913, 12 2012.
- MANJUNATHA, P.; VERMA, A. K.; SRIVIDYA, A. Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method. In: **2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems**. Kharagpur, India: [s.n.], 2008. p. 1–6.
- MANRIQUE, J. A.; RUEDA-RUEDA, J. S.; PORTOCARRERO, J. M. T. Contrasting internet of things and wireless sensor network from a conceptual overview. In: **2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)**. Chengdu, China: [s.n.], 2016. p. 252–257.
- MARCELLINO, M.; STOCK, J. H.; WATSON, M. W. A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. **Journal of Econometrics**, v. 135, n. 1, p. 499 – 526, 2006.
- MASOUM, A.; MERATNIA, N.; HAVINGA, P. J. A distributed compressive sensing technique for data gathering in wireless sensor networks. **Procedia Computer Science**, v. 21, p. 207 – 216, 2013. The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH).
- MATIN, M.; ISLAM, M. Overview of wireless sensor network. In: **Wireless Sensor Networks - Technology and Protocols**. [S.I.]: InTech, 2012.
- MATUSIAK, E.; ELDAR, Y. C. Sub-nyquist sampling of short pulses. **IEEE Transactions on Signal Processing**, IEEE, v. 60, n. 3, p. 1134–1148, 2012.
- MEKKI, K.; BAJIC, E.; CHAXEL, F.; MEYER, F. A comparative study of ipwan technologies for large-scale iot deployment. **ICT Express**, 2018. ISSN 2405-9595.
- MHATRE, V.; ROSENBERG, C. Design guidelines for wireless sensor networks: communication, clustering and aggregation. **Ad Hoc Networks**, Elsevier BV, v. 2, n. 1, p. 45–63, jan 2004.

- MONTEIRO, L. C.; DELICATO, F. C.; PIRMEZ, L.; PIRES, P. F.; MICELI, C. Dpcas: Data prediction with cubic adaptive sampling for wireless sensor networks. In: AU, M. H. A.; CASTIGLIONE, A.; CHOO, K.-K. R.; PALMIERI, F.; LI, K.-C. (Ed.). **Green, Pervasive, and Cloud Computing**. Cham: Springer International Publishing, 2017. p. 353–368.
- MORETTIN, P. A.; TOLOI, C. M. de C. **Modelos para previsão de séries temporais**. São Paulo, SP: Instituto de matematica pura e aplicada, 1981. v. 1.
- MOTA, J. F. C.; XAVIER, J. M. F.; AGUIAR, P. M. Q.; PÜSCHEL, M. Basis pursuit in sensor networks. In: **2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. Prague, Czech Republic: [s.n.], 2011. p. 2916–2919.
- MUSZNICKI, B.; ZWIERZYKOWSKI, P. Survey of simulators for wireless sensor networks. **International Journal of Grid and Distributed Computing**, Citeseer, v. 5, n. 3, p. 23–50, 2012.
- NAGELKERKE, N. J. A note on a general definition of the coefficient of determination. **Biometrika**, Oxford University Press, v. 78, n. 3, p. 691–692, 1991.
- NATRELLA, M. Nist/sematech e-handbook of statistical methods. NIST/SEMATECH, 2010.
- NETWORKS, U. P. **SmartMeter Energy Consumption Data in London Households**. 2015. <<https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>>. [Online; acessado em 08/03/2018].
- OLAH, C. **Understanding LSTM Networks**. 2014. <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. [Online; acessado em 06/12/2017].
- OLIVEIRA, S.; KNIESS, J. Abordagem baseada em análise de dispersão para redução de dados em redes de sensores sem fio. In: **Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**. Gramado, RS: SBC, 2019.
- OLIVEIRA, S.; KNIESS, J.; MARQUES, V. A method based on dispersion analysis for data reduction in wsn. In: **Proceedings of International Conference on Green, Pervasive, and Cloud Computing 2019**. Uberlândia, MG: Springer International Publishing, 2019.
- OLIVEIRA, S.; KNIESS, J.; PARPINELLI, R.; CASTAÑEDA, W. A. Predição de séries temporais em internet das coisas com redes neurais recorrentes. In: **Anais do 50º Simpósio de Pesquisa Operacional**. Rio de Janeiro, RJ: [s.n.], 2018.
- POOR, H. V. Signal processing issues in wireless sensor networks. In: **2006 14th European Signal Processing Conference**. Florence, Italy: [s.n.], 2006. p. 1–1.
- POURGHEBLEH, B.; NAVIMIPOUR, N. J. Data aggregation mechanisms in the internet of things: A systematic review of the literature and recommendations for future research. **Journal of Network and Computer Applications**, Elsevier BV, v. 97, p. 23–34, nov 2017.
- PULE, M.; YAHYA, A.; CHUMA, J. Wireless sensor networks: A survey on monitoring water quality. **Journal of Applied Research and Technology**, v. 15, n. 6, p. 562 – 570, 2017. ISSN 1665-6423.

QAISAR, S.; BILAL, R. M.; IQBAL, W.; NAUREEN, M.; LEE, S. Compressive sensing: From theory to applications, a survey. **Journal of Communications and Networks**, v. 15, n. 5, p. 443–456, Oct 2013. ISSN 1229-2370.

QUEENSLAND GOVERNMENT. **Ambient estuarine water quality monitoring data (includes near real-time sites) - 2012 to present day**. 2015. <<https://data.qld.gov.au/dataset/ambient-estuarine-water-quality-monitoring-data-near-real-time-sites-2012-to-present-day>>. [Online; acessado em 08/03/2018].

QUER, G.; ZORDAN, D.; MASIERO, R.; ZORZI, M.; ROSSI, M. Wsn-control: Signal reconstruction through compressive sensing in wireless sensor networks. In: **IEEE Local Computer Network Conference**. Denver, CO, USA: [s.n.], 2010. p. 921–928.

RAJU, T. N. William sealy gosset and william a. silverman: two “students” of science. **Pediatrics**, Am Acad Pediatrics, v. 116, n. 3, p. 732–735, 2005.

RAO, S. S.; REDDY, N. C. K. An energy efficient improved RPL routing protocol. In: **2017 International Conference on Intelligent Computing and Control (I2C2)**. [S.I.]: IEEE, 2017.

RASTEGARNIA, A.; SOLOUK, V. Performance evaluation of castalia wireless sensor network simulator. In: **2011 34th International Conference on Telecommunications and Signal Processing (TSP)**. Budapest, Hungary: [s.n.], 2011. p. 111–115.

RAUT, R. P.; GAHARWAR, A. S. S.; JADE, S. L.; BAIS, N. N.; VIRSEN, A. S.; SO-NEKAR, S. A survey on data aggregation mechanism in wireless sensor networks. **International Journal of Scientific Research in Computer Science, Engineering and Information Technology**, v. 3, p. 90–96, 2 2018.

RAZALI, N. M.; WAH, Y. B. et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. **Journal of statistical modeling and analytics**, v. 2, n. 1, p. 21–33, 2011.

REINA, D. G.; TORAL, S. L.; BARRERO, F.; BESSIS, N.; ASIMAKOPOULOU, E. The role of ad hoc networks in the internet of things: A case scenario for smart environments. In: **Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence**. [S.I.]: Springer Berlin Heidelberg, 2013. p. 89–113.

ROSNER, B.; GLYNN, R. J.; LEE, M.-L. T. The wilcoxon signed rank test for paired comparisons of clustered data. **Biometrics**, Wiley Online Library, v. 62, n. 1, p. 185–192, 2006.

RUBIO, G.; POMARES, H.; HERRERA, L. J.; ROJAS, I. Kernel methods applied to time series forecasting. In: **Computational and Ambient Intelligence**. [S.I.]: Springer Berlin Heidelberg, 2007. p. 782–789.

RÜPING, S. **SVM kernels for time series analysis**. [S.I.], 2001.

SANTINI, S.; ROMER, K. An adaptive strategy for quality-based data reduction in wireless sensor networks. In: **Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)**. [S.I.: s.n.], 2006. p. 29–36.

SANTOS, B. P.; SILVA, L.; CELES, C.; BORGES, J. B.; NETO, B. S. P.; VIEIRA, M. A. M.; VIEIRA, L. F. M.; GOUSSEVSKAIA, O. N.; LOUREIRO, A. Internet das coisas: da teoriaa prática. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, 2016.

SAPANKEVYCH, N. I.; SANKAR, R. Time series prediction using support vector machines: A survey. **IEEE Computational Intelligence Magazine**, v. 4, n. 2, p. 24–38, May 2009. ISSN 1556-603X.

SCIKIT-LEARN. **Support Vector Regression (SVR) using linear and non-linear kernels**. 2018. <http://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html>. [Online; acessado em 26/10/2018].

SCOTTA, F. C. Análise de tendências em séries temporais de dados meteorológicos e dados de sensoriamento remoto orbital em áreas de vegetação campestre natural do bioma pampa, localizadas na depressão central do rs. 2012.

SHAHRASBI, B.; TALARI, A.; RAHNAVARD, N. Tc-csbp: Compressive sensing for time-correlated data based on belief propagation. In: **2011 45th Annual Conference on Information Sciences and Systems**. [S.I.: s.n.], 2011. p. 1–6.

SHANTHAMALLU, U. S.; SPANIAS, A.; TEPEDELENLIOGLU, C.; STANLEY, M. A brief survey of machine learning methods and their sensor and iot applications. In: **2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)**. [S.I.: s.n.], 2017. p. 1–8.

SHARP, M. **Neural Network Programming with Python: Create Your Own Neural Network!** CreateSpace Independent Publishing Platform, 2016. ISBN 9781539381952. Disponível em: <<https://books.google.com.br/books?id=GX4fvgAACAAJ>>.

SHI-NASH, A.; HARDOON, D. R. DATA ANALYTICS AND PREDICTIVE ANALYTICS IN THE ERA OF BIG DATA. In: **Internet of Things and Data Analytics Handbook**. John Wiley & Sons, Inc., 2016. p. 329–345. Disponível em: <<https://doi.org/10.1002/9781119173601.ch19>>.

SHNAYDER, V.; HEMPSTEAD, M.; CHEN, B.-r.; ALLEN, G. W.; WELSH, M. Simulating the power consumption of large-scale sensor network applications. In: **Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems**. New York, NY, USA: ACM, 2004. (SenSys '04), p. 188–200. ISBN 1-58113-879-2.

SILVA, S. da. **Detecção de danos estruturais usando analise de series temporais e atuadores e sensores piezeletricos**. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, SP, 2008.

SILVER, N. **O sinal e o ruído**. [S.I.]: Editora Intrínseca, 2013.

SIRSIKAR, S.; ANAVATTI, S. Issues of data aggregation methods in wireless sensor network: A survey. **Procedia Computer Science**, v. 49, p. 194 – 201, 2015. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15).

- SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. **Statistics and computing**, Springer, v. 14, n. 3, p. 199–222, 2004.
- SONI, S. K.; CHAND, N.; SINGH, D. P. Reducing the data transmission in wsns using time series prediction model. In: **2012 IEEE International Conference on Signal Processing, Computing and Control**. [S.I.: s.n.], 2012. p. 1–5.
- SOUA, R.; MINET, P. A survey on energy efficient techniques in wireless sensor networks. In: **2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)**. [S.I.]: IEEE, 2011.
- SPANDAN, G.; PATEL, A.; MANJUNATH, C.; NAGARAJ, G. Data aggregation protocols in wireless sensor networks. **International journal of computational engineering research**, v. 3, n. 5, p. 18–24, 2013.
- STALLINGS, W. Ieee 802.11: Wireless lans from a to n. **IT professional**, IEEE, n. 5, p. 32–37, 2004.
- TAN, L.; WU, M. Data reduction in wireless sensor networks: A hierarchical lms prediction approach. **IEEE Sensors Journal**, v. 16, n. 6, p. 1708–1715, March 2016. ISSN 1530-437X.
- TAYEH, G. B.; MAKHOUL, A.; DEMERJIAN, J.; LAIYMANI, D. A new autonomous data transmission reduction method for wireless sensors networks. In: **2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)**. [S.I.: s.n.], 2018. p. 1–6.
- TROPP, J. A.; GILBERT, A. C. Signal recovery from random measurements via orthogonal matching pursuit. **IEEE Transactions on Information Theory**, v. 53, n. 12, p. 4655–4666, Dec 2007. ISSN 0018-9448.
- TZAGKARAKIS, G.; STARCK, J. L.; TSAKALIDES, P. Joint sparse signal ensemble reconstruction in a wsn using decentralized bayesian matching pursuit. In: **2011 19th European Signal Processing Conference**. [S.I.: s.n.], 2011. p. 338–342. ISSN 2076-1465.
- UNIVERSITÉ LIBRE DE BRUXELLES. **Wireless Sensor Networks Laboratory**. 2012. <http://di.ulb.ac.be/lab0/>. [Online; acessado em 14/02/2019].
- VAPNIK, V. N. **The Nature of Statistical Learning Theory**. Berlin, Heidelberg: Springer-Verlag, 1995. ISBN 0-387-94559-8.
- VARGA, A.; HORNIG, R. An overview of the omnet++ simulation environment. In: **Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops**. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008. (Simutools '08), p. 60:1–60:10.
- VC, S.; K, J.; KUMAR, S. M. Energy minimization by removing data redundancy in wireless sensor networks. In: **2014 International Conference on Communication and Signal Processing**. IEEE, 2014. Disponível em: <https://doi.org/10.1109/iccsp.2014.6950130>.

VERHEGGHEN, A.; MAYAUX, P.; WASSEIGE, C. D.; DEFOURNY, P. Mapping congo basin vegetation types from 300 m and 1 km multi-sensor time series for carbon stocks and forest areas estimation. **Biogeosciences**, Copernicus GmbH, v. 9, n. 12, p. 5061, 2012.

VILLAS, L.; BOUKERCHE, A.; ARAUJO, R. B. de; LOUREIRO, A. A. F. Highly dynamic routing protocol for data aggregation in sensor networks. In: **The IEEE symposium on Computers and Communications**. [S.l.: s.n.], 2010. p. 496–502. ISSN 1530-1346.

VILLAS, L. A.; BOUKERCHE, A.; ARAUJO, R. B.; LOUREIRO, A. A. A reliable and data aggregation aware routing protocol for wireless sensor networks. In: **Proceedings of the 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems**. New York, NY, USA: ACM, 2009. (MSWiM '09), p. 245–252.

VITO, S. D.; MASSERA, E.; PIGA, M.; MARTINOTTO, L.; FRANCIA, G. D. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. **Sensors and Actuators B: Chemical**, Elsevier BV, v. 129, n. 2, p. 750–757, feb 2008.

WANT, R. Near field communication. **IEEE Pervasive Computing**, Institute of Electrical and Electronics Engineers (IEEE), v. 10, n. 3, p. 4–7, 2011. Disponível em: <<https://doi.org/10.1109/mprv.2011.55>>.

WILMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. **Climate research**, JSTOR, v. 30, n. 1, p. 79–82, 2005.

WRIGHT, D. J. Forecasting data published at irregular time intervals using an extension of. **Manage. Sci.**, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 32, n. 4, p. 499–510, abr. 1986.

XIE, R.; JIA, X. Transmission-efficient clustering method for wireless sensor networks using compressive sensing. **IEEE Transactions on Parallel and Distributed Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 25, n. 3, p. 806–815, mar 2014. Disponível em: <<https://doi.org/10.1109/tpds.2013.90>>.

XU, X.; ANSARI, R.; KHOKHAR, A.; VASILAKOS, A. V. Hierarchical data aggregation using compressive sensing (hdacs) in wsns. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 11, n. 3, p. 45:1–45:25, fev. 2015. ISSN 1550-4859. Disponível em: <<http://doi.acm.org/10.1145/2700264>>.

YANG, J.; WANG, Y.; ZHANG, D. Prediction revision strategies for data transmission in wireless sensor networks. In: **2008 8th IEEE International Conference on Computer and Information Technology**. [S.l.: s.n.], 2008. p. 827–832.

YETGIN, H.; CHEUNG, K. T. K.; EL-HAJJAR, M.; HANZO, L. A survey of network lifetime maximization techniques in wireless sensor networks. **IEEE Communications Surveys & Tutorials**, Institute of Electrical and Electronics Engineers (IEEE), v. 19, n. 2, p. 828–854, 2017. Disponível em: <<https://doi.org/10.1109/comst.2017.2650979>>.

- YI, K.; FENG, R.; YU, N.; CHEN, P. Pared: A testbed with parallel reprogramming and multi-channel debugging for wsns. In: **2013 IEEE Wireless Communications and Networking Conference (WCNC)**. [S.I.: s.n.], 2013. p. 4630–4635. ISSN 1525-3511.
- YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Comput. Netw.**, Elsevier North-Holland, Inc., New York, NY, USA, v. 52, n. 12, p. 2292–2330, ago. 2008.
- ZANJIREEH, M. M.; SHAHRABI, A.; LARIJANI, H. ANCH: A new clustering algorithm for wireless sensor networks. In: **2013 27th International Conference on Advanced Information Networking and Applications Workshops**. IEEE, 2013. Disponível em: <<https://doi.org/10.1109/waina.2013.242>>.
- ZECHINELLI-MARTINI, J. L.; BUCCIOL, P.; VARGAS-SOLAR, G. Energy aware data aggregation in wireless sensor networks. In: **2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)**. Chennai, India: [s.n.], 2011. p. 1–5.
- ZEGARRA, E. T.; SCHOUERY, R. C. S.; MIYAZAWA, F. K.; VILLAS, L. A. A continuous enhancement routing solution aware of data aggregation for wireless sensor networks. In: **2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)**. [S.I.: s.n.], 2016. p. 93–100.
- ZHAI, X.; JING, H.; VLADIMIROVA, T. Multi-sensor data fusion in wireless sensor networks for planetary exploration. In: **2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)**. Leicester, UK: [s.n.], 2014. p. 188–195.
- ZHANG, Z.; LAI, C.; CHAO, H. A green data transmission mechanism for wireless multimedia sensor networks using information fusion. **IEEE Wireless Communications**, v. 21, n. 4, p. 14–19, August 2014. ISSN 1536-1284.
- ZHENG, H.; LI, J.; FENG, X.; GUO, W.; CHEN, Z.; XIONG, N. Spatial-temporal data collection with compressive sensing in mobile sensor networks. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 17, n. 11, p. 2575, 2017.
- ZHENG, H.; XIAO, S.; WANG, X.; TIAN, X.; GUIZANI, M. Capacity and delay analysis for data gathering with compressive sensing in wireless sensor networks. **IEEE Transactions on Wireless Communications**, v. 12, n. 2, p. 917–927, February 2013. ISSN 1536-1276.
- ZOU, P.; LIU, Y. An efficient data fusion approach for event detection in heterogeneous wireless sensor networks. **Applied Mathematics & Information Sciences**, Natural Sciences Publishing Corp, v. 9, n. 1, p. 517, 2015.