Notas em Matemática Aplicada e-ISSN 2236-5915

Volume 60, 2012

Editores

Cassio Machiaveli Oishi
Universidade Estadual Paulista - UNESP
Presidente Prudente, SP, Brasil

Fernando Rodrigo Rafaeli
Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Rosana Sueli da Motta Jafelice (Editor Chefe)
Universidade Federal de Uberlândia - UFU
Uberlândia, MG, Brasil

Rubens de Figueiredo Camargo
Universidade Estadual Paulista - UNESP
Bauru, SP, Brasil

Sezimária de Fátima P. Saramago
Universidade Federal de Uberlândia - UFU
Uberlândia, MG, Brasil

Vanessa Avansini Botta Pirani (Editor Adjunto)
Universidade Estadual Paulista - UNESP
Presidente Prudente, SP, Brasil

SBMAC **Sociedade Brasileira de Matemática Aplicada e Computacional**

2012

A Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC publica, desde as primeiras edições do evento, monografias dos cursos que são ministrados nos CNMAC.

Para a comemoração dos 25 anos da SBMAC, que ocorreu durante o XXVI CNMAC em 2003, foi criada a série **Notas em Matemática Aplicada** para publicar as monografias dos minicursos ministrados nos CNMAC, o que permaneceu até o XXXIII CNMAC em 2010.

A partir de 2011, a série passa a publicar, também, livros nas áreas de interesse da SBMAC. Os autores que submeterem textos à série Notas em Matemática Aplicada devem estar cientes de que poderão ser convidados a ministrarem minicursos nos eventos patrocinados pela SBMAC, em especial nos CNMAC, sobre assunto a que se refere o texto.

O livro deve ser preparado em **Latex (compatível com o Miktex versão 2.7), as figuras em eps** e deve ter entre **80 e 150 páginas**. O texto deve ser redigido de forma clara, acompanhado de uma excelente revisão bibliográfica e de **exercícios de verificação de aprendizagem** ao final de cada capítulo.

Veja todos os títulos publicados nesta série na página
http://www.sbmac.org.br/notas.php

Sociedade Brasileira de Matemática Aplicada e Computacional

2012

REDUÇÃO DE DADOS EM REDES DE SENSORES

Andre L. L. Aquino
alla@ic.ufal.br

Laboratório de Computação Científica e Análise Numérica
Instituto de Computação
Universidade Federal de Alagoas

Paulo R. S. Silva Filho
progerio@ic.uff.br

Laboratório MídiaCom
Instituto de Computação
Universidade Federal Fluminense

SBMAC **Sociedade Brasileira de Matemática Aplicada e Computacional**

São Carlos - SP, Brasil
2012

Coordenação Editorial: Elbert Einstein Nehrer Macau

Coordenação Editorial da Série: Rosana Sueli da Motta Jafelice

Editora: SBMAC

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Catalogação elaborada pela Biblioteca do IBILCE/UNESP
Bibliotecária: Maria Luiza Fernandes Jardim Froner

Aquino, Andre L. L.
Redução de Dados em Redes de Sensores - São Carlos, SP : SBMAC, 2012, 112 p., 20.5 cm - (Notas em Matemática Aplicada; v. 60)

e-ISBN 978-85-8215-011-5

1. Redes de sensores 2. Redução de dados
I. Aquino, Andre L. L. II. Silva Filho, Paulo R. S. III. Título. IV. Série

CDD - 51

Conteúdo

Prefácio

As redes de sensores sem fio têm emergido como um dos grandes desafios de pesquisa da última década e apresentam soluções de monitoramento, coleta de dados e auxílio a outras ferramentas de controle de ambientes diversos. Entretanto, devido às diversas limitações de processamento, armazenamento de dados, consumo de energia e tempo de resposta dos nós sensores, muitas técnicas são estudadas de forma a otimizar o desempenho e a resposta dessas redes, sendo uma delas a redução de dados. Redes de sensores sem fio é uma área de pesquisa interdisciplinar. Dentre as diversas áreas destacamos a simulação e estatítica aplicada, por intermédio de aplicações de redução de dados univariada ou multivariada. Nesse contexto, a estatística torna-se uma ferramenta essencial, por exemplo, na proposição de técnicas de amostragem univariadas e multivariadas. Ao escrever o presente livro, o objetivo principal é oferecer ao estudante de graduação, e mesmo de pós-graduação, da área de ciências exatas, um texto introdutório sobre redução de dados em redes de sensores, destacando um modelo arquitetônico para projetar tais aplicações, o uso das técnicas de amostragem univariadas baseada em wavelets e multivariadas basedas em PCA. Além disso, apresenta-se a utilização da arquitetura de redução em algumas aplicações de redes de sensores. Ao término de cada capítulo há uma pequena lista de exercícios. Uma extensa bibliografia sobre o assunto foi anexada ao final do texto.

Andre L. L. Aquino
Paulo R. S. Silva Filho

Capítulo 1

Introdução

Neste capítulo é apresentada uma breve introdução a respeito das redes de sensores sem fio (RSSFs), seguida da apresentação da motivação para utilizarmos redução de dados nessas redes. Após isso, são apresentadas as diferentes granularidades de aplicações em RSSFs onde a redução de dados pode ser utilizada de forma satisfatória. O capítulo é finalizado com uma lista de exercícios de verificação de aprendizagem.

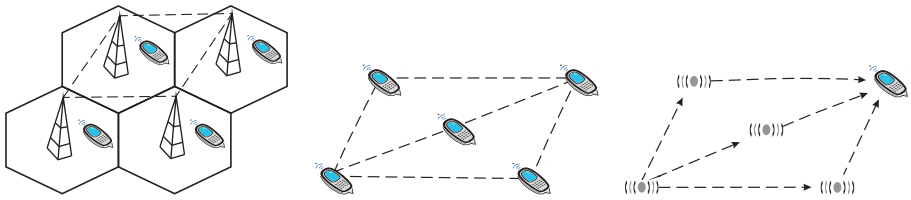
1.1 Redes de sensores sem fio

Os diferentes fenômenos encontrados na natureza podem ser descritos por algumas grandezas, tais como temperatura, pressão e umidade; que podem ser monitoradas por dispositivos com capacidade de sensoriamento, processamento e comunicação. O conjunto desses dispositivos, que trabalham de forma independente, organizada e cooperativa, é conhecido como uma rede de sensores sem fio (RSSF) [2, 15, 26, 77]. Além dos nós, essas redes também podem ser formadas por elementos atuadores que interferem no meio monitorado, um ou mais sorvedouros que recebem e processam os dados, além dos *gateways* que são responsáveis pela comunicação da RSSF com outras redes.

Os estudos focados nessas redes cresceram bastante na última década, graças à quantidade de cenários e situações onde as RSSFs podem ser aplicadas, dentre as quais podemos listar [2, 25, 26, 51, 63, 82]:

- Monitoramento e coleta de dados acerca das condições climáticas em florestas;
- Monitoramento das condições estruturais de construções históricas e antigas;
- Medição dos batimentos cardíacos, nível de oxigenação do sangue, temperatura corporal e pressão arterial de pacientes à distância;
- Rastreamento e monitoramento de tropas inimigas em áreas de guerras;
- Análise das condições do microclima em ambientes de computação de alto desempenho (*clusters* ou supercomputadores).

Inicialmente, para melhor contextualizar as RSSFs no ambiente sem fio, consideremos as redes estruturadas e *ad hoc*. Com relação às redes estruturadas, temos que elas possuem nós subordinados à estação base, que é responsável pela comunicação entre os elementos da rede (Figura 1.1(a)). Já as redes *ad hoc* [66] não utilizam uma estação base para prover a comunicação entre os elementos da rede, pois a comunicação é feita utilizando os nós que estão entre a origem e o destino (Figura 1.1(b)). Assim, as RSSFs possuem uma forma de comunicação similar às redes *ad hoc*, com o objetivo de propagar os dados sensoriados para um elemento externo à rede (Figura 1.1(c)).



(a) Rede estruturada. (b) Rede não estruturada. (c) Rede de sensores sem fio.

Figura 1.1: Tipos de redes sem fio.

Os nós sensores coletam dados do ambiente e os processam localmente ou de forma cooperativa entre nós vizinhos. Ao final, a informação processada pode ser enviada para o usuário. Devido ao seu

tamanho, os nós sensores possuem uma arquitetura simples e com limitações de processamento e armazenamento, sendo formados pelos componentes básicos ilustrado na figura 1.2.

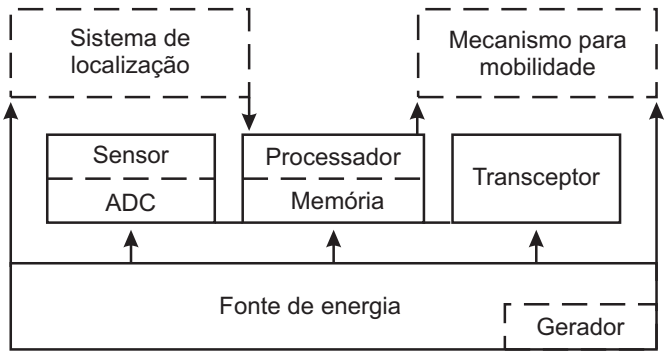


Figura 1.2: Estrutura do nó sensor com os quatro componentes principais e os três componentes opcionais (em linhas tracejadas).

Os componentes básicos são:

- *Unidade perceptiva* que pode possuir alguns sensores e um conversor de sinais analógicos para digitais (ADC);
- *Unidade de processamento* com memória e processador;
- *Transceptor* para permitir a comunicação com outros nós sensores;
- *Fonte de energia* que geralmente não é renovável.

De forma opcional, podem existir elementos que complementam a estrutura dos sensores, como *sistemas de localização*, *mecanismos de mobilidade* e/ou *geradores de energia*. Além dos nós sensores, uma RSSF pode ter outros três tipos de nós:

- *Nós atuadores* que possuem a função de atuar ou interferir no meio onde estão inseridos a fim de corrigir falhas e/ou controlar o objeto monitorado;

- *Nós sorvedouros* ou nós de monitoração que recebem os dados e os processam de forma a extrair alguma informação útil para o usuário;
- *Nós gateways* que são responsáveis por prover a comunicação da RSSF com outras redes de computadores.

Esses três elementos básicos, bem como a estrutura típica de uma RSSF, são ilustrados na figura 1.3. É importante destacar que esses elementos não precisam ser fisicamente distintos, por exemplo, o sorvedouro e o *gateway* podem ser o mesmo dispositivo.

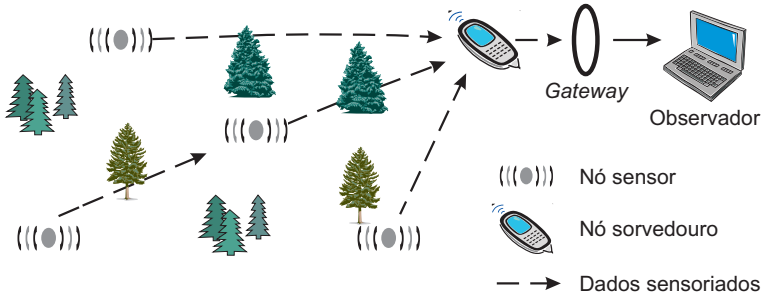


Figura 1.3: Estrutura de uma RSSF considerando não apenas o nó sensor como também os demais elementos básicos.

Ao observarmos as RSSFs considerando os seus elementos básicos, a forma com que os nós são dispostos numa área de sensoriamento e a forma com que os fenômenos são monitorados, pode-se fazer uma distinção entre os diferentes tipos de RSSFs existentes. Com isso, as RSSFs podem ser classificadas como segue [77]:

- *Hierárquica*, se ela possui agrupamentos de nós onde existe um líder que representa cada agrupamento, caso contrário a rede é considerada *plana*;
- *Homogênea*, se os nós possuem a mesma configuração de *hardware*, caso contrário a rede é considerada *heterogênea*;
- *Simétrica*, se todos os nós possuem o mesmo raio de comunicação, caso contrário a rede é considerada *assimétrica*;

- *Contínua*, se os dados coletados são enviados continuamente ou *programada*, se os dados são enviados obedecendo a programação previamente estabelecida;
- *Dirigida a eventos*, se a rede envia dados apenas quando ocorre algum evento ou *sob demanda*, quando a rede permite a consulta parcial ou total dos dados aos nós.

1.2 Redução de dados em redes de sensores de sem fio

Cada nó sensor tem a capacidade de monitorar um ou mais fenômenos. Os dados que representam esses fenômenos monitorados podem ser classificados como segue:

- *Dados univariados* que representam um único conjunto de valores de um mesmo fenômeno. Por exemplo, os dados monitorados por um nó que possui apenas um sensor de temperatura.
- *Dados multivariados* que representam mais de um conjunto de valores de um mesmo fenômeno ou de vários fenômenos. Por exemplo, os dados recebidos por um nó responsável por processar os dados monitorados por um conjunto de nós que possuem apenas um sensor de temperatura, ou os dados monitorados por um nó que possui simultaneamente os sensores de temperatura, pressão e umidade.

Os fenômenos monitorados são reportados através de uma comunicação sem fio *ad hoc* [66] para o sorvedouro. Essa comunicação, devido às características da aplicação, possui restrições de energia, processamento de dados, tempo de resposta e largura de banda. Especificamente, no que diz respeito à largura de banda, enviar grandes quantidades de dados pode ser problemático devido à quantidade de nós que terão que acessar o meio, podendo causar atraso demasiado no tempo de resposta e, assim, invalidando os dados. Além disso, um grande tráfego na rede degrada rapidamente seu tempo de vida

(tempo que um nó sensor permanece funcionando antes de sua bateria se esgotar).

Devido à essas restrições, é necessário adotar alguma estratégia para o tratamento dos dados a fim de reduzir ou selecionar apenas os dados mais relevantes que representam o fenômeno monitorado. Dentre as diversas abordagens para redução de dados em RSSFs sem fio pode-se destacar:

- *Agregação de dados* que efetua a redução dos dados sensoriados seguindo alguma métrica exigida pela aplicação. Tem como objetivo principal diminuir o tráfego na rede, independente da qualidade dos dados reduzidos [47].
- *Amostragem adaptativa* que, ao longo do tempo de vida da rede, modi-fica a forma de sensoriamento com o objetivo de propagar apenas a informação mais relevante para a aplicação. Caso os dados possuam características distintas essa técnica apresentará um nível de redução baixo [6, 14, 20].
- *Amostragem de dados multivariados* que utiliza métodos para estimar o comportamento dos dados multivariados, como sua correlação, permitindo que apenas as diferenças na correlação dos dados observadas ao longo do tempo sejam propagadas até o sorvedouro [43, 44].

De forma geral a necessidade de se reduzir os dados nas RSSFs parte do seguinte problema geral:

> *Dado que as aplicações em RSSFs necessitam reduzir os dados, o projetista estará interessado em efetuar a redução de tal forma que seja possível reduzir gastos de energia e atraso de pacotes na rede. Além disso, a aplicação precisa manter sua representatividade, onde as decisões tomadas após a redução sejam equivalentes às decisões que seriam tomadas sem a redução.*

Esse problema geral guia toda a motivação de se efetuar redução de dados em RSSFs e será fruto de nosso estudo durante do este livro.

1.3 Hierarquia das aplicações em redes de sensores sem fio

As RSSFs podem ser utilizadas em uma grande variedade de aplicações. Essas aplicações podem ter um caráter de monitoramento, onde apenas dados do ambiente são coletados; ou um caráter de atuação, onde ocorre intervenção no meio monitorado [12, 13]. De forma geral, podemos considerar três níveis de granularidade nas aplicações em RSSFs: sensoriamento, mecanismos de infraestrutura e aplicações para o usuário.

1.3.1 Aplicações de sensoriamento

As RSSFs possuem restrições de recursos que, aliadas às necessidades das aplicações, tornam o projeto dessas redes complexo. Nesse contexto, existem diversas linhas de pesquisa que tratam problemas relacionados com o projeto dessas redes, como a auto-organização [19, 28, 72] e o gerenciamento de recursos [32, 70, 87, 88].

Por tratarem de um tipo específico de redes *ad hoc* e serem utilizadas em ambientes hostis com condições imprevisíveis, as RSSFs devem ser autoconfiguráveis, adaptáveis e possuir um gerenciamento escalável. Devido às características da aplicação de sensoriamento, as RSSFs possuem um modelo centrado nos dados [40], pois o objetivo dessas redes é levar a informação sensoriada para um ponto fora da rede. Essa característica permite a integração das operações da camada de sensoriamento com a camada de rede, oferecendo soluções mais eficientes.

Fatores relacionados às características da rede, tipos e configurações dos sensores influenciam diretamente no desenvolvimento das aplicações de sensoriamento. Considerando essas características, pode-se classificar as aplicações de sensoriamento como segue:

- *Aplicação de monitoramento*, onde os dados são enviados periodicamente ou em resposta a um evento inesperado. Nesse caso, o nó sensor faz apenas um pré-processamento nos dados deixando as operações mais elaboradas para serem executadas em outros

elementos da rede com maior poder de processamento. Esse pré-processamento é necessário, pois o grande volume de dados sensoriados, se enviados sem nenhum tratamento, pode consumir a energia dos nós (diminuindo o tempo de vida dos mesmos) e comprometer os objetivos da rede.

- *Aplicações baseadas em consultas*, onde os dados são enviados apenas quando requisitados por algum elemento externo à rede. Nesse caso, o nó sensor deve executar algum processamento sobre os dados, de tal forma que apenas o resultado desse processamento seja guardado para ser enviado quando solicitado. Isso ocorre, pois o armazenamento de todos os dados sensoriados pode ser muito caro para o nó, devido a limitação de armazenamento e, principalmente, se as consultas não forem frequentes.

Para o tratamento dos dados nas aplicações de monitoramento podemos utilizar técnicas de redução como agregação de dados [47], fusão de dados [59] ou *stream* de dados [6]. Para as aplicações de consultas, geralmente a rede é vista como um grande banco de dados onde operações sobre os dados são calculadas internamente [1, 53].

1.3.2 Mecanismos de infraestrutura

As RSSFs possuem as seguintes funcionalidades básicas [51]:

- *Estabelecimento* da rede, que consiste na configuração inicial da rede;
- *Manutenção* que consiste na adaptação da rede às mudanças de configurações que surgem ao longo do tempo;
- *Sensoriamento* que trata da coleta de dados sobre o ambiente;
- *Processamento* dos dados a serem enviados para o sorvedouro;
- *Comunicação* que é responsável pelo envio desses dados.

Discutiremos de forma mais detalhada apenas as funcionalidades de estabelecimento e manutenção, mais especificamente a tarefa de

roteamento onde efetivamente podemos efetuar tarefas de redução de dados.

O estabelecimento de uma RSSF basicamente envolve as atividades de disposição dos nós na área a ser monitorada e formação da rede. Essa fase ocorre antes do sensoriamento, assim, os nós podem realizar tarefas de controle de densidade, formação de agrupamentos e montagem da estrutura de roteamento. Após o estabelecimento da rede é necessário manter a estrutura funcionando eficientemente durante todo o tempo de vida da rede. Segundo Loureiro et al. [51]: "O objetivo da manutenção é prolongar o tempo de vida da rede, reduzir a imprevisibilidade e atender aos requisitos da aplicação, pois ao longo do tempo alguns nós atingem níveis de energia que podem restringir de forma parcial ou total sua capacidade". Todas as tarefas realizadas para o estabelecimento da rede devem ser repetidas durante a manutenção, seja periodicamente ou na ocorrência de um determinado evento. Essa decisão dependerá do objetivo da aplicação.

Uma das tarefas que é considerada tanto na fase de estabelecimento como na fase de manutenção é a montagem da estrutura de roteamento. Uma abordagem bastante utilizada em RSSFs para essa tarefa é o roteamento baseado em árvore, cuja montagem consiste em configurar os nós da rede para que eles saibam para qual vizinho enviar suas informações sensoriadas [28, 60].

Um algoritmo de roteamento baseado em árvore é composto pelas seguintes fases:

- *Construção da árvore* baseada em alguns requisitos de rede ou da aplicação. É construída via inundação[1] do sorvedouro para os nós. É nesse momento que as informações da aplicação são passadas para os nós sensores.
- *Encaminhamento* os dados sensoriados pelos nós fontes são encaminhados para o sorvedouro através da estrutura em árvore.
- *Reconstrução da árvore* em alguns casos é necessário reconstruir a árvore, pois a topologia da rede pode mudar por falha, desligamento ou esgotamento da energia dos nós. A estratégia de

[1] O termo inundação é mais conhecido do inglês *flooding*.

reconstrução pode ser feita de forma pró-ativa ou reativa, dependendo do gerenciamento da rede.

Como as RSSFs são centradas nos dados, é possível que a informação presente nos dados seja importante nas decisões da camada de roteamento. Caso a rede tenha restrições de energia e atraso, a identificação de dados redundantes na camada de roteamento pode habilitar reduções ou descarte desses dados, ou ainda, caminhos de roteamento alternativos podem ser utilizados para entregar dados com maior prioridade.

1.3.3 Aplicações para o usuário

As aplicações para o usuário em RSSFs, normalmente, apenas utilizam a infraestrutura da rede para obter informações do fenômeno monitorado. Contudo, existem aplicações as quais o simples envio das informações sensoriadas não é suficiente e aspectos relacionados ao tempo de resposta são fundamentais [17, 52]. Alguns exemplos dessas aplicações com exigência de prazos são:

- *Aplicações militares* que necessitam efetuar a coleta dos dados e a atuação no ambiente monitorado em tempo real;
- *Aplicações de segurança* que utilizam sensores acústicos e de vídeo para detectar movimentos e soar algum alarme num intervalo de tempo bem pequeno;
- *Aplicações para detecção de ataques biológicos* que utilizam sensores para identificar a presença de elementos biológicos no corpo humano ou no ambiente.

Em sistemas embutidos de tempo real tradicionais, o prazo da tarefa é um ponto crítico a ser considerado (tempo real *hard*). Algoritmos de escalonamento são desenvolvidos para reduzir ou evitar a perda dos prazos, seja estatisticamente ou dinamicamente. Em um ambiente dinâmico, o mecanismo de controle de admissão aceitará ou rejeitará a tarefa baseado na restrição de tempo e de outros recursos do sistema. O projeto dessas aplicações é mais complexo, pois é

concebido para ambientes específicos [17]. Em RSSFs é comum haver aplicações de tempo real *soft*, pois o ambiente não é controlado. A aplicação normalmente usa métodos probabilísticos para tratar os dados e não tem confirmação na comunicação. Esses aspectos tornam o uso de tempo real *hard* em RSSFs bem mais difícil. Por convenção, utilizaremos o termo "aplicações de tempo real" ao invés de tempo real *soft* em RSSFs.

Considerando as aplicações de tempo real em RSSFs, utilizar uma solução que garanta *a priori* o atendimento dos prazos é bem mais difícil, como dito acima, devido às características dessas redes. No entanto, podemos utilizar soluções aproximadas que identificam, durante o roteamento, o momento em que os dados não podem ser entregues a tempo, exigindo que algum processamento nos dados seja feito, de tal forma que alguma informação útil possa chegar ao usuário dentro dos prazos exigidos. No projeto de um sistema de tempo real para uma RSSF, nós devemos conhecer o comportamento do atraso do envio dos dados para cada solução de uma dada aplicação e, com isso, aplicar a melhor solução de processamento dos dados para atender as exigências requeridas.

1.4 Exercícios de verificação de aprendizagem

1. Defina os principais elementos de uma RSSF.
2. Quais as diferenças entre uma rede estruturada, uma rede *ad hoc* e uma RSSFs?
3. Defina as classificações de RSSFs apresentadas nesse capítulo.
4. Quais as fases que compõem um algoritmo de roteamento baseado em árvore?
5. Cite três exemplos de pontos críticos nas aplicações de RSSFs.
6. Diferencie tempo real *hard* de tempo real *soft*.
7. Apresente um exemplo de aplicação para cada item da questão anterior.

8. Quais são as funcionalidades básicas de uma RSSFs?

9. Suponha que uma aplicação que deseje monitorar pontos de queimadas em uma reserva florestal na Amazônia. Apresente os pontos chaves para a implementação de uma RSSF nessa floresta (arquitetura, aspectos críticos dessa rede, disposição dos nós etc.) que atenda aos requisitos dessa aplicação, sabendo que os sensores devem monitorar a temperatura e a umidade relativa do ar e devem enviar os dados a uma estação base a cada 10 minutos.

10. Além das aplicações mencionadas nesse capítulo, apresente outro tipo de aplicação ao qual as RSSFs podem empregadas, apresentando os detalhes de arquitetura e os aspectos críticos.

Capítulo 2

Fundamentação teórica e abordagens usuais para redução de dados

Neste capítulo, apresentaremos os principais conceitos utilizados para as diferentes técnicas de redução de dados abordadas nesse livro. Falaremos sobre transformadas de *Wavelets*, análise de componentes e testes estatísticos utilizados para a análise da robustez da redução. Em seguida, mostraremos alguns trabalhos situando sua importância para a área. Por fim, finalizaremos o capítulo com uma lista de exercícios de verificação de aprendizagem.

2.1 Wavelets background

Esta seção introdutória sobre *Wavelets* considera apenas os aspectos inerentes para um bom o entendimento da aplicação de *Wavelets* em redução de dados. Para um estudo mais aprofundado sugerimos a leitura do material de Mallat [55].

2.1.1 A transformada

Sejam V_j' uma sequência de espaços fechados $L(\mathbb{R}^2)$ e $f(x) \in L(\mathbb{R}^2)$ um sinal observado. Cada V_j' representa uma aproximação do sinal original, considerando uma resolução de 2^j. Uma resolução suave ocorre quando o menor valor possível para j é usado.

Os detalhes da projeção entre 2^j e 2^{j-1}, denotado por W_j, é definido com

$$W_j \oplus V_j' = V_{j-1}',$$

onde $\oplus$ denota a doma de dois espaços vetorias. Com isso, V_j' pode ser decomposto por intermédio da soma direta desses subespaços,

$$V_j' = W_{j+1} \oplus W_{j+2} \oplus \cdots \oplus W_J \oplus V_J',$$

onde $j < J$ e todos os subespaços são ortogonais.

A transformada discreta de *Wavelets* é utilizada para a análise de dados discretos. Filtros discretos são definidos para escolher os níveis de frequência nos dados que varia em uma escala temporal. Dois conjuntos de funções são aplicadas: as funções de escala $\phi(t)$ e as funções de Wavelets $\psi(t)$. Ambas estão relacionadas ao menor e maior filtro passante, chamadas de vetores *Wavelets* e escala, respectivamente.

Assim, $f(x)$ pode ser aproximada pela seguinte expansão

$$f(x) = \sum_n s_{i_0}[n]\psi_{i_0,n}(x) + \sum_{i=i_0}^{i_1} w_i[n]\,\phi_{i,n}(x)$$

onde $s_{i_0}[n] \in V_J'$ são os coeficientes de escala,

$$s_{i_0}[n] = \int f(x)\,\psi_{i_0,n}(x)\,dx$$

e $w_i[n] \in W_i$ são os coeficientes de *Wavelets*,

$$w_i[n] = \int f(x)\,\phi_{i,n}(x)\,dx$$

2.1.2 Propriedades

A transformada de *Wavelets* de uma função $f(x)$ é uma função bidimensional $\gamma(s,\tau)$. As variáveis s e τ são as novas dimensões, escala e translação, respectivamente.

A transformada de *Wavelets* é composta pelas funções $\phi(t)$ e $\psi(t)$ com $\int \psi(t)dt = 0$. Esta condição afirma que *Wavelets* tem um suporte compacto, uma vez que a maior parte do seu valor é restrito a um intervalo finito, o que significa dizer que ele tem valor zero fora deste intervalo. Além disso, uma versão expandida de $\psi(t)$ é $\psi_{i,k}(t) = 2^{\frac{i}{2}}\psi(2^i - k)$. Estes casos caracterizam a propriedade de localização espacial da transformada de *Wavelets*.

Outra propriedade é a suavidade da transformada de *Wavelets*. Considere a expansão de $\gamma(s,\tau)$, em $\tau = 0$, em uma série de Taylor de ordem n,

$$\gamma(s,0) = \frac{1}{\sqrt{s}}\left[\sum_{p=0}^{n} f^{(p)}(0)\int \frac{t^p}{p!}\psi(t/s)dt + O(n+1)\right],$$

onde $f^{(p)}$ é a p^{th} derivada de f e $O(n+1)$ representa o resto da expansão. Um *Wavelets*, $\psi(t)$, tem L momentos nulos se

$$\int_{-\infty}^{\infty} t^k\psi(t)dt = 0,$$

para $0 \leq k < L$. Se

$$M_k = \int t^k\psi(t)dt,$$

nós temos

$$\gamma(s,0) = \frac{1}{\sqrt{s}}\left[\frac{f^{(0)}(0)}{0!}M_0 s^1 + \cdots + \frac{f^{(n)}(0)}{n!}M_n s^{n+1} + O(s^{n+2})\right]$$

Os momentos de fuga da função são

$$M_n(0,l) = 0, \text{ for } l = 0,1,...,L-1$$

onde L é o número de momento do *Wavelets*. Da condição de admissão, nós temos que o momento 0^{th} (M_0) é igual a zero. Se os outros

momentos M_n são zero, então $\gamma(s,\tau)$ irá convergir para uma função suave $f(t)$. Portanto, se $f(t)$ é descrita por uma função polinomial de grau maior que $(L-1)$, o termo $O(s^{n+2})$ será zero e pequenos valores aparecerão como uma combinação linear de ψ em $\gamma(s,\tau)$.

O grau de regularidade da *Wavelets* e a sua taxa de decaimento é governada pelo número de momentos nulos que apresenta. Esta propriedade é importante para deduzir as propriedades de aproximação exibidas pela *Wavelets* nos espaços de multirresolução. Quando uma *Wavelets* possui vários momentos nulos resulta em coeficientes de menores valores, porque os coeficientes de *Wavelets* das escalas mais finas de uma função $f(x)$ são essencialmente nulos onde $f(x)$ é suave.

2.1.3 Base de Haar

A base de Haar possui a seguinte função *Wavelets*

$$\psi(x) = \begin{cases} 1, & \texttt{se } x \in [0, \frac{1}{2}) \\ -1, & \texttt{se } x \in [\frac{1}{2}, 1) \\ 0, & \texttt{caso contrário} \end{cases}$$

e a função escala ϕ

$$\phi(t) = \begin{cases} 1, & t \in [0,1) \\ 0, & t \notin [0,1) \end{cases}$$

A função ϕ dada pela função característica no intervalo $[0,1)$ é chamada de função escala associada as *Wavelets* de Haar. A base de Haar é a única que possui um suporte compacto e uma fórmula analítica fechada.

2.1.4 Base de Daubechies

As *Wavelets* de Daubechies apresentam uma capacidade de análise e síntese muito mais efetiva do que as de Haar por possuírem maior regularidade (suavidade) e aproximarem melhor funções (suaves) em

$L2(R)$. Elas não possuem fórmula analítica fechada, sendo necessário calcular numericamente seus coeficientes.

As *Wavelets* de Daubechies são numeradas em função do número de momentos nulos que possuem. As funções ψ são parametrizadas por um inteiro $N > 1$. Tais *Wavelets* e suas funções escala possuem suporte compacto em intervalos de tamanho $2N - 1$. A partir de operações de translação este intervalo se torna $[-N+1, N]$. A base de Daubechies possuem todos os momentos de ordem zero até ordem $N-1$ como nulos, onde $\int_{\infty}^{\infty} x^l \phi(x)dx$, para $l = 0, 1, \cdots, N-1$. Com isso, maior o valor escolhido para N, mais suave será a base de Daubechies. Para $N = 2$, temos a 2_ϕ também chamada de *Daub4*.

2.1.5 Base de Coiflets

As *Wavelets* periódicas são aplicadas aos intervalos limitados das funções. Para aplicar a transformada periódica é necessário considerar que os limites da função alvo sejam repetidos. Para evitar essa condição a base de Coiflets possui a propriedade dos momentos nulos em sua função escala. A base de Coiflets é uma extensão a Daubechies, mas sua função escala também possui momentos nulos, de forma que

$$\begin{cases} \int \psi(x)dx = 1 \\ \int x^l \psi(x)dx = 0, \quad l = 0, 1, ..., L-1 \\ \int x^l \phi(x)dx = 0, \quad l = 0, 1, ..., L-1 \end{cases}$$

Com isso, nesta base, tanto ψ quanto ϕ possuem momentos nulos. Com isso, a função ϕ é mais suave e simétrica do que a ϕ da base de Daubechies, possuindo característica de interpolação que a torna melhor para aproximações de funções polinomiais. Considerando $2L$ como a quantidade de momentos da Coiflets e uma função $f(x)$ no intervalo $[p, q]$ temos:

$$\int_p^q f(x)\phi(x) = \int_p^q f(0) + f'(0) + \cdots + \frac{f^{2L-1}(0)x^{2L-1}}{(2L-1)!} + \cdots \approx f(0)$$

Considerando que $f(x)$ seja um polinômio de ordem $p \leq 2L-1$, então existe um polinômio $g(x)$ de mesmo grau tal que

$$f(x) \approx \sum_{\tau} g(\tau)\phi_{s,\tau}(x).$$

Esta propriedade da base de coiflets permite que durante a aplicação da transformada, alguns termos podem ser calculados diretamente ou amostrados considerando um erro de aproximação, dado por

$$\left\| f(x) - \sum_{\tau} g(\tau)\phi_{s,\tau}(x) \right\| = O(2^{s.2L}),$$

uma vez que somente no caso que existam infinitos momentos nulos teríamos a propriedade $\int f(x)\delta(x)dx = f(0)$, conhecida como propriedade de Dirac δ. Essa propriedade faz com que os coeficientes em $\gamma(s,\tau)$ sejam uma representação esparsa da função $f(x)$ sendo necessário apenas uma pequena quantidade de coeficientes para aproximar $f(x)$, tal que

$$\gamma(s,\tau) = f(2^{-(s)}\tau) + O(2^{s.2L}).$$

Considerando a L-ésima ordem da base de Coiflets, é possível definir um rápido algoritmo, usando a propriedade de menor erro, quando a função $f(t)$ é uma função de suavização. Cada elemento em $V'_j[t]$ pode ser aproximado por

$$V'_j[t] = 2^{-\frac{i}{2}} f(2^i t)$$

e cada coeficiente de *Wavelets* aproximado ($W_t[t]$ na escala 2^i), são computados como

$$W_j[t] = \sum g[2t-n]V'_{j+1}[t].$$

2.2 Análise de componentes

Nesta seção discorre-se a respeito de algumas técnicas baseadas em análise de componentes, apresentando a base para um melhor entendimento das mesmas. Essa descrição é necessária, uma vez que alguns algoritmos de amostragem multivariados se beneficiam das técnicas de análise de componentes.

2.2.1 Componentes principais

Análise de Componentes Principais[1] [35, 41, 48, 62], também conhecida como transformação de Karhunen-Loève, é uma das ferramentas mais poderosas para o tratamento de dados multivariados. É uma transformação entre espaços γ-dimensionais, derivada da matriz de covariância dos dados de entrada gerando um novo conjunto de dados, de modo que cada valor resultante é uma combinação linear dos valores originais. Essas combinações lineares são denominadas componentes principais. O número de componentes principais é igual ao número de dimensões dos dados originais e esses podem ser ordenados de acordo com a sua variância. Com isso, a primeira componente principal explica a maior parte da variância total da matriz de entrada, ou seja, é aquela que provê mais informações. Por outro lado, a última componente principal tem a menor variância, ou seja, menor quantidade de informações.

Após a determinação das componentes principais, os valores numéricos ou escores de cada uma dessas componentes podem ser calculados e posteriormente analisados, através de técnicas estatísticas tais como análise de variância, análise de regressão, entre outras. Em geral, a utilização de PCA se dá para reduzir a quantidade de dimensões dos dados a serem avaliadas. Dessa forma, entre as s componentes principais calculadas, são escolhidas k componentes, onde $k < s$. Essas k componentes são escolhidas de forma a explicarem a maior parte da variância global dos dados originais, permitindo uma análise sem perdas significativas de qualidade. Além disso, segundo Mingoti [58]:

> É comum utilizar os escores das componentes principais para condução de análise estatística de dados ou para a simples ordenação (*ranking*) dos elementos amostrais observados, com o intuito de identificar aqueles que estão com maiores, ou menores, valores globais das componentes.

A propriedade mais importante do novo conjunto de dados gerado por PCA, ou seja, das componentes principais, é que essas compo-

[1] Análise de componentes principais é abreviada na literatura como PCA do inglês *Principal Component Analysis*

nentes são não-correlacionadas entre si [41], garantindo dessa forma que não haja redundância entre os dados e que seja obtido um novo conjunto de dados com propriedades para análise multivariada. Dessa forma, a técnica consegue reduzir a dimensão dos dados mantendo a qualidade dos mesmos. A transformação de componentes principais pode ser descrita nas seguintes etapas:

1. Calcular Σ, a matriz de covariância dos dados (supõe-se que ela seja definida positiva pois trata-se de variâncias).

2. Decompor Σ nos autovetores U e autovalores λ. Essa matriz será diagonalizável, uma vez que a matriz de covariância é definida positiva [49].

3. Calcular o novo conjunto de dados, multiplicando o valor de cada variável pela matriz dos autovetores.

Os autovalores representam o comprimento dos eixos das componentes principais do conjunto de dados e são medidos na unidade da variância. Associado a cada autovalor, existe um vetor de módulo unitário chamado autovetor. Os elementos de cada autovetor são fatores de ponderação que definem a contribuição da variável da matriz de dados original para uma componente principal, numa combinação linear. Os autovetores representam as direções dos eixos das componentes principais.

Considere que V representa um conjunto de dados multivariados sensoriados. Assim, o método de componentes principais pode ser formulado da seguinte forma: dada uma matriz de dados originais V, com s variáveis correlacionadas, aplicar PCA consiste em calcular a matriz C, que possui s variáveis não correlacionadas, de forma que cada componente principal será calculada por

$$C_i = u_i'[V - \overline{V}], \tag{2.2.1}$$

onde para cada $1 \leq i \leq s$, $u_i = (u_{i,1}, \ldots, u_{i,s})$ é o autovetor i da matriz de covariância dos dados V.

Outra propriedade importante do PCA é que a equação (2.2.1) pode ser invertida, restaurando as variáveis originais em função das

componentes principais. Para isso utiliza-se

$$V = \overline{V} + UC. \tag{2.2.2}$$

Devido ao autovetor U ser ortonormal, temos $U^{-1} = U'$; com isto, dada a matriz C, os dados originais V podem ser unicamente determinados pela equação (2.2.2).

PCA pode apresentar deficiências no que se refere à robustez do método quando da presença de valores discrepantes ou atípicos nos dados de entrada. Para tentar resolver esse problema de robustez do método, existe a técnica PCA-robusta [24, 75, 81]. A principal diferença dessa técnica para a técnica PCA tradicional está na maneira de se calcular a matriz de covariância Σ dos dados, que no caso de PCA-robusta pode ser feita calculando-se cada elemento de Σ separadamente, através de estimadores robustos específicos para o coeficiente de correlação ou covariância, ou ainda, estimando Σ como um todo.

2.2.2 Componentes independentes

Outra técnica utilizada para o tratamento de dados multivariados é denominada Análise de Componentes Independentes[2] (ICA) [21, 37, 38]. O objetivo dessa técnica é encontrar uma transformação na qual as componentes C_i são estatisticamente independentes umas das outras. De acordo com Hyvarinen [37], ICA pode ser considerada uma técnica para reduzir redundância. A definição mais simples e mais utilizada por pesquisadores considera que a transformação linear através de ICA consiste em estimar o seguinte modelo geral para os dados

$$V = A\,C,$$

onde V são os dados de entrada, A é uma "matriz de mistura" retangular e C_i no vetor $C = (C_1, \ldots, C_n)^T$, assim como no caso de PCA representam as componentes, que nesse caso, diferentemente de PCA, são consideradas independentes. Assume-se ainda que C possui média zero e covariância finita.

[2] Análise de Componentes Independentes é abreviada na literatura como ICA do inglês *Independent Component Analysis*

Algumas restrições devem ser consideradas para se assegurar a identidade do modelo ICA [37]. A primeira restrição considera que apenas uma das componentes independentes C_i pode ser gaussiana, todas as outras devem ser não gaussianas. Isso porque, para variáveis aleatórias gaussianas, a simples inexistência de correlação implica em independência, e dessa forma, qualquer representação não-correlacionada resultaria em componentes independentes. Contudo, se mais de uma das componentes são gaussianas, ainda existe a possibilidade de se identificar as componentes independentes não gaussianas, bem como as correspondentes colunas da "matriz de misturas". Outra restrição é que todas as colunas da matriz A devem ser linearmente independentes.

2.2.3 Componentes independentes versus componentes principais

ICA possui algumas semelhanças e diferenças em relação à PCA. Assim como PCA, ICA tem o objetivo de realizar uma transformação linear em um conjunto de dados. As diferenças começam pelas considerações contraditórias em relação à gaussianidade, uma vez que PCA assume que dados são gaussianos, enquanto ICA assume que são não gaussianos. Além disso, a aplicação principal de PCA é para a redução de dimensionalidade, enquanto ICA pode reduzir, aumentar ou manter o número de dimensões constante.

Conforme descrito por Hyvarinen [37], diferentemente de PCA, a definição de ICA não implica em uma ordenação das componentes independentes em relação à variância dos dados. Embora seja possível determinar a ordem das componentes, por exemplo, de acordo com a não gaussianidade das componentes, usualmente considera-se que as componentes têm a mesma variância. Dessa forma, neste trabalho assume-se que a primeira componente independente possui a maior variância.

Outro ponto importante da técnica ICA é que, de forma semelhante à PCA, em cada componente independente, é possível determinar os escores dessas componentes e, a partir desses escores, realizar análises estatísticas dos dados. Para a determinação desses escores das compo-

nentes independentes será utilizado o algoritmo FastICA [36, 39]. Este algoritmo apresenta diversas vantagens em relação aos outros métodos baseados em ICA, tais como:

- Sua convergência é cúbica (ou pelo menos quadrática), enquanto os demais apresentam convergência linear, ou seja, a convergência do FastICA é mais rápida.
- O algoritmo é mais fácil de usar, pois não há necessidade de escolher parâmetros de tamanho de passo como nos algoritmos baseados em gradiente.
- FastICA encontra diretamente as componentes independentes de praticamente qualquer distribuição não gaussiana, usando alguma não linearidade $\boldsymbol{g}$.
- A escolha de uma não linearidade $\boldsymbol{g}$ adequada pode otimizar seu desempenho.
- É possível estimar as componentes uma a uma.
- FastICA herda vantagens de algoritmos neurais: é paralelo, distribuído, computacionalmente simples e requer pequeno espaço de memória.

2.3 Testes estatísticos

A seção que segue, apresentamos alguns testes estatísticos úteis para a análise da qualidade dos dados após a redução. Tais testes são importantes para identificarmos a precisão de cada técnica de redução utilizada.

2.3.1 Análise univariada

Considerando dados univariados, duas análises serão apresentadas: a aproximação entre duas distribuições de frequência; e o valor absoluto do erro relativo.

Para a avaliação da aproximação entre as distribuições de frequência temos o *Kolmogorov-Smirnov* (teste KS) [64, 74]. Esse teste avalia se duas amostras V e V' têm distribuições similares, não exigindo que as amostras sigam a distribuição normal, ou seja, caso os valores amostrados sigam outra distribuição este teste também pode ser utilizado. O teste KS é descrito a seguir:

1. Construir a distribuição acumulada F_n dos dois grupos V e V' usando a mesma classe para ambas as distribuições.

2. Determinar as diferenças acumuladas para cada ponto da distribuição e considerar a maior das diferenças (D_{max}).

3. Computar o valor crítico,

$$D_{crit} = y\sqrt{(|V| + |V'|)/|V|\,|V'|}$$

   onde y é um valor tabulado e representa o nível de significância do teste.

4. As amostras seguem a mesma distribuição se

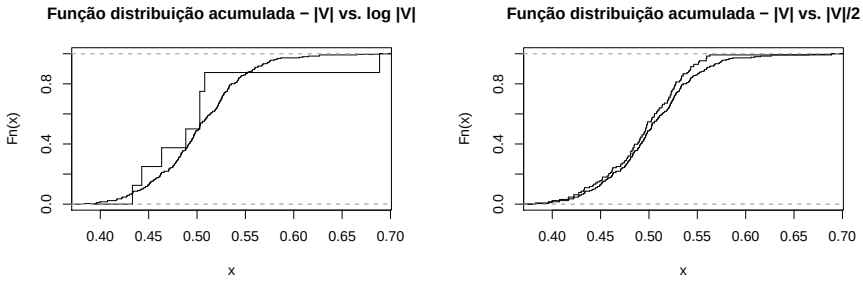
$$D_{max} \leq D_{crit}.$$

Apenas como ilustração, considere a figura 2.1 que apresenta a comparação entre as distribuições de frequência acumulada, com $|V| = 256$ e $|V'| = \{\log|V|, |V|/2\}$[3] onde $V' \subset V$. Em ambos os casos, através do teste KS, temos que V' segue a mesma distribuição de V.

Como o teste KS apenas identifica se duas amostras seguem a mesma distribuição, é importante avaliar se os conjuntos V e V' possuem a média de seus valores próximos. Para isso podemos calcular a maior distância entre $\overline{V}$ e os valores do intervalo de confiança $IC = [v_{inf}; v_{sup}]$ de $\overline{V}$. Os passos para essa avaliação são descritos a seguir:

1. Obter a média dos valores dos dados reduzidos e originais, que são respectivamente $\overline{V}$ e $\overline{V'}$.

[3]Em todo o trabalho, ao utilizarmos $\log x$, estaremos sempre nos referindo ao logaritmo de x na base dois.



(a) Comparando com log dos dados. (b) Comparando com a metade dos dados.

Figura 2.1: Função da distribuição acumulada para 256 valores.

2. Calcular o intervalo de confiança IC com confiança de 95% para $\overline{V'}$.

3. Calcular o valor absoluto da maior diferença entre $\overline{V}$ e IC

$$\epsilon = \max\{|v_{inf} - \overline{V}|, |v_{sup} - \overline{V}|\}.$$

2.3.2 Análise multivariada

Para a análise de dados multivariados consideramos os seguintes testes: o teste de hipótese *Analysis of Variance - ANOVA* [76] e o valor absoluto do erro relativo aplicado a cada variável, para determinar a qualidade dos dados reduzidos.

O teste de hipótese *ANOVA* tem por objetivo avaliar se existem diferenças estatisticamente significativas entre as médias do conjunto de dados original e do conjunto de dados reduzido. O cálculo é dado por,

$$F = D_B^2 / D_W^2,$$

onde D_B^2 representa a dispersão entre os conjuntos V e V' e D_W^2 a dispersão dentro dos conjuntos. A partir desse cálculo, o $p - valor$ é utilizado para decidir se a hipótese nula H_0 deve ser aceita ou rejeitada. Nesse caso, aceitar a hipótese nula indica que não existem diferenças significativas entre as variâncias dos dois conjuntos. Resultados para

o $p-valor$ acima de $0,05$ são considerados satisfatórios para aceitação da hipótese nula.

O valor absoluto do erro relativo considera a comparação entre as médias dos dados originais V e reduzidos V'. Esse erro é dado por,

$$\epsilon = 100\, Max\{\forall_i \left|(\overline{V_i} - \overline{V_i'})/\overline{V_i}\right|\}.$$

Note que ϵ é calculado para cada sensor i e somente o maior deles será utilizado na estatística da qualidade dos dados.

2.4 Abordagens clássicas para redução de dados em redes de sensores sem fio

Devido às restrições das RSSFs, diversos trabalhos têm sido propostos para reduzir a quantidade de dados que trafegam nessas redes, visando economizar energia e, consequentemente, aumentar o tempo de vida da rede. Para efetuar essa redução diversas técnicas têm sido empregadas, dentre as quais destacam-se a agregação de dados, amostragem adaptativa, redução baseada em *stream* de dados e a redução de dados multivariados.

2.4.1 Redução de dados univariados

A seguir apresentaremos alguns trabalhos relacionados com redução de dados univariados.

Agregação de dados

A agregação é uma das técnicas mais empregadas para redução de dados univariados em RSSFs. Em [47], os autores avaliam o impacto da agregação de dados em RSSFs. Conforme apresentado, a utilização de esquemas de agregação ótimos é uma tarefa NP-Difícil, tornando necessária a aplicação de esquemas subótimos para realizar esse processo, tais como CNS (*Center at Nearest Source*), SPT (*Shortest Paths Tree*) e GIT (*Greedy Incremental Tree*). Segundo os autores, a agregação possui ganhos significativos em condições específicas, por exemplo,

quando há um grande número de nós fonte e estes estão próximos entre si e a vários saltos do sorvedouro. Além disso, ela proporciona um aumento do atraso, que no pior caso será proporcional à distância entre o sorvedouro e a fonte mais distante. Outro fator importante a ser considerado é que seu desempenho é influenciado também pelo número de fontes, pela topologia e pela densidade da rede, o que inviabiliza sua utilização em uma grande gama de aplicações em RSSFs.

No trabalho de Dasgupta et al. [23], é apresentado um esquema de seleção de árvores de agregação de dados visando maximizar o tempo de vida da rede. Os autores fazem uma adaptação do algoritmo MLDA (Maximum Lifetime Data Aggregation) [45], uma vez que sua complexidade o torna inviável para RSSFs de grande porte. O algoritmo utiliza uma seleção inteligente de árvores de agregação, a partir de um conjunto de árvores de agregação candidatas. A abordagem considera uma rede hierárquica, onde os dados coletados pelos nós em cada cluster são agregados e transmitidos a uma estação base através de um nó líder, que é escolhido de uma forma *round-robin*. Nas primeiras n transmissões, o algoritmo utiliza n árvores de agregação, e a partir daí, nas próximas transmissões, essas árvores são selecionadas também em uma maneira *round-robin*. Resultados demonstraram a eficiência da técnica quanto à extensão do tempo de vida da rede. Fator interessante nesse trabalho é a utilização de redes hierárquicas, onde o nó líder é o responsável por efetuar a agregação, o que pode também ser empregado para a redução de dados multivariados.

Em [90], é proposto um algoritmo para realizar a coleta de informações de forma adaptável aos recursos, visando alcançar a qualidade de serviço desejada e satisfazer os requisitos de recursos e latência nas tarefas de coleta e disseminação de dados em RSSFs. O método realiza agregação de dados local, considerando o compromisso entre latência, eficiência de energia e qualidade de acordo com os recursos e requisitos específicos da aplicação, reduzindo a quantidade de informações transmitidas na rede. Todos os nós podem transmitir dados em nome de outros e de acordo com uma probabilidade, processam e agregam os dados. Assim, quando uma estação base solicita uma tarefa, ela informa a todos os nós as restrições a serem consideradas. Cada nó

da rede decide se participa ou não da tarefa, e aqueles que participam, de acordo com seu nível de energia e o tempo necessário para entrega dos dados, agregam os dados ou simplesmente retransmitem sem realizar a agregação. Para contornar o problema do atraso, existe um número máximo de nós que realizam a agregação, escolhidos de acordo com uma função de probabilidade e considerando o tempo para realizar o processo. Segundo os autores, os resultados mostraram que a agregação diminui a quantidade de dados transmitidos e economiza energia, e a técnica consegue ainda ganhos com relação ao atraso inserido na rede. Entretanto, o modelo utilizado nesse trabalho é apenas estatístico, além de sua conclusão ser muito superficial, o que aponta para uma necessidade de avaliar a técnica mais profundamente e em cenários próximos dos reais.

Em [86], é proposto um algoritmo baseado em árvore de menor caminho, com energia máxima, para agregação de dados em RSSFs, o MESPT (*Maximum-Energy Short Path Tree*). O algoritmo possui duas fases, sendo que na primeira ele constrói uma árvore de caminho de energia máxima, que balanceia o consumo de energia entre os nós. Na segunda fase, a árvore é reestruturada através do algoritmo de menor caminho, para tentar minimizar a latência na agregação dos dados. A rede é modelada como um grafo onde cada vértice representa um nó e uma aresta entre dois vértices indica que dois nós podem se comunicar diretamente. Cada aresta possui um peso, que indica a distância entre os dois nós e cada nó tem ciência da sua energia residual e da sua localização. A energia máxima de um caminho é dada pelo menor valor de energia residual entre os nós nesse caminho. O caminho de energia máxima é aquele que possui maior valor de energia máxima entre todos os caminhos possíveis entre dois nós. O objetivo da solução proposta é encontrar uma árvore de agregação abrangendo todos os sensores na área monitorada e encontrar uma raiz adequada para a coleta de dados. Dessa forma, a árvore pode minimizar a latência na agregação e balancear o consumo de energia, maximizando o tempo de vida da rede. O interesse então, é encontrar o menor caminho, com a maior quantidade de energia residual. Uma vez que a raiz, além de coletar dados, é responsável também por estabelecer a rota até o

sorvedouro, o nó com maior energia residual é escolhido como raiz. A agregação executa uma fusão na rede dos pacotes de dados vindos de diferentes sensores para a raiz, com o intuito de minimizar o número de transmissões e o tamanho dos dados, e, consequentemente, economizar energia. Essa agregação pode ser executada quando os dados são altamente correlacionados e o trabalho considera ainda que um nó intermediário pode agregar vários pacotes recebidos em um único pacote a ser enviado. A avaliação do algoritmo é feita através de uma comparação entre o MESPT e o *Directed Diffusion* [20], que é um protocolo de roteamento para coleta de dados baseado nos atributos desses dados e que suporta agregação se esses atributos forem relevantes. O objetivo é analisar o tempo de vida da rede e o atraso no envio dos pacotes. De acordo com os resultados apresentados, em ambos os critérios, MESPT se mostrou mais eficiente que o protocolo DD. Contudo, apesar de citar que a agregação pode ser executada quando os dados são altamente correlacionados, não é feita nenhuma descrição sobre a avaliação dessa correlação. Outro ponto importante é que não há nenhuma análise sobre a qualidade dos dados após efetuada a agregação. Dependendo da aplicação, esse fator pode ser de fundamental importância.

Amostragem adaptativa

Em [56], é apresentado um modelo geral dos mecanismos de amostragem adaptativa em RSSFs. Essas redes precisam ser autoconfiguráveis e a grande quantidade de dispositivos e a característica dinâmica dos ambientes são desafios para se obter essa autonomia. Quanto maior a variação no ambiente monitorado, menor tende a ser a taxa de precisão das informações. Devido a esses fatores, um mecanismo de controle é utilizado em cada sensor, o que faz com que a taxa de sensoriamento seja dinâmica e adaptativa. Esse mecanismo trabalha com um modelo do ambiente a ser monitorado e de acordo com esse modelo um número maior ou menor de coletas será feito. A quantidade de amostras e a complexidade do modelo influenciam no uso de recursos como memória e energia, criando uma necessidade de se obter um compromisso entre a taxa de amostragem e a precisão das informações.

Uma abordagem para a utilização de amostras adaptativas é apresentada por Willett et al. [79] – o Backcasting. A técnica possui dois passos para reduzir o consumo de energia e manter a precisão das informações. No primeiro passo, uma estimativa inicial do ambiente é formada usando um subconjunto dos nós sensores. Esse subconjunto envia as informações para um centro de fusão, que, com base nessa estimativa, envia mensagem para que outros sensores sejam ativados, afim de melhorar a precisão das informações, o que ocorre no segundo passo. A ideia principal é que com a estimativa inicial seja possível conhecer as correlações no ambiente, podendo assim, determinar o número de sensores que não precisam ser ativados. Com isso, a amostragem adaptativa pode diminuir a quantidade de comunicações e, consequentemente, economizar energia. Resultados apresentados indicam que a solução proposta consegue reduzir o consumo de energia, mantendo um nível de precisão das informações adequado. Esses resultados, no entanto, são estatísticos, não havendo simulações do comportamento real da rede. Outro fator relevante é que uma variação maior nos dados coletados pelos sensores pode afetar drasticamente o desempenho da técnica, uma vez que o número de sensores que serão ativados pelo centro de fusão pode ser muito grande.

Uma das formas de amostragem é a utilização de esquemas de predição dos valores coletados tanto no nó fonte como no sorvedouro, onde apenas as informações que desviam dessa predição são transmitidas. Santini & Romer [71] apontam os problemas dessa técnica, e propõem uma solução para esses problemas. A técnica de predição, apesar de reduzir o consumo de energia na rede, precisa ter um conhecimento *a priori* das informações, fazendo com que o modelo tenha que ser constantemente atualizado, aumentando, assim, os custos da comunicação. Os autores propõem um algoritmo que possui um esquema de predição adaptativa não baseado em um modelo de conhecimento prévio. No método proposto, são usados filtros de predição idênticos no nó fonte e no sorvedouro. Quando o nó coleta os dados e envia para o sorvedouro, ambos aplicam o algoritmo de predição e fazem uma estimativa de erro, que é comparada com um limite previamente definido. Assim, o nó enviará informações, agora, somente se esse li-

mite de erro for excedido. Resultados apresentados mostram que a técnica é muito eficiente. Contudo, o trabalho considerou apenas ambientes que necessitam de entrega contínua de dados em intervalos de tempo regulares, ou seja, as irregularidades temporais não foram consideradas. Além disso, considerou que o enlace de comunicação é livre de perdas e possíveis falhas nos nós não foram consideradas, sendo bastante diferente das condições reais encontradas por essas redes.

No entanto, conforme Genesan et al. [30], é muito importante considerar as irregularidades espaço-temporais no processo de amostragem em RSSFs, o que não é considerado na maioria dos trabalhos. Os fenômenos não são distribuídos uniformemente no espaço (os recursos dos sensores variam para executar o sensoriamento; terrenos não são uniformes), e amostragem temporal regular requer relógios sincronizados em todos os pontos de mensuração, aumentando o custo de transmissão e o gasto de energia, o que inviabiliza sua aplicação. Segundo os autores, esquemas de agregação geram resultados imprecisos, a eficiência da compressão é bastante reduzida e esquemas de armazenamento aumentam o *overhead* no roteamento. Propostas para solução desses problemas são apresentadas, tais como a utilização de interpolação de dados e segmentação do sinal temporal seguido pelo alinhamento, para resolver o problema da agregação e o uso de virtualização e detecção de fronteira para reduzir custos de roteamento e armazenamento. Os resultados obtidos não são significativos e precisam ser melhor analisados, mas fica clara a importância de se considerar essas irregularidades encontradas nas RSSFs

De acordo com Wu & Luo [80], os esquemas existentes de agendamento do período de dormência dos nós assumem que todos os nós da rede possuem a mesma taxa de transmissão, ou o mesmo número de transmissões por unidade de tempo, e atribuem um ciclo ativo-dormindo fixo para cada nó, de acordo com a taxa de transmissão. Todavia, esses esquemas podem causar falhas na transmissão em RSSFs que utilizam amostras adaptativas. Uma das formas de resolver esse problema seria especificar uma taxa de transmissão única para todos os nós, mas é difícil escolher um valor para essa taxa, o que pode ocasionar problemas na transmissão e no consumo de energia.

Para resolver esse problema, é proposto um esquema de agendamento na camada de rede, o SPAS (*Scheduling Protocol for Adaptive Sampling*). Nesse esquema, diferentemente do que ocorre na camada MAC, o agendamento requer informações sobre a taxa de transmissão, que está mais facilmente disponível na camada de rede do que na MAC. Além disso, o agendamento na camada de rede permite uma otimização entre roteamento e agendamento. Outra diferença é que, embora o agendamento na camada MAC possa ser mais eficiente em relação ao consumo de energia, na camada de rede ele é mais flexível, pois não tem requisitos estritos em *slots* de tempo. Isso traz vantagens para a amostragem adaptativa, uma vez que as taxas de transmissão podem ser alteradas com frequência. No método proposto, cada nó conhece o tempo de transmissão de seus vizinhos, podendo com isso, enviar e receber dados dos mesmos, quando necessário. Após a transmissão dos dados, o nó é colocado em estado de dormência, até que termine o ciclo ativo-dormindo, para economizar energia. Assim, como a transmissão dos dados é a operação final em cada ciclo, o fato do nó dormir não afeta outras operações. Outra característica do SPAS é a otimização do roteamento, permitindo a um nó escolher, dinamicamente, um vizinho mais próximo ao sorvedouro. Essa otimização faz com que o tempo em que o nó fica ativo seja reduzido, bem como o tempo para acordar. A avaliação do método se dá em um simulador implementado pelos autores, onde se avaliou o consumo de energia médio dos nós e a taxa de perda de pacotes, que nesse caso representa a qualidade dos dados. Resultados apresentados indicam que o método é eficiente, tanto em termos do consumo de energia, quanto da qualidade dos dados, ou seja, da taxa de perda de pacotes. Entretanto, não são apresentadas no trabalho as características principais do simulador implementado, o que dificulta uma avaliação mais aprofundada ou uma comparação com outros métodos existentes.

Redução baseada em stream de dados

Outro método utilizado para efetuar a redução de dados em RSSFs é a redução baseada em *stream* de dados. No trabalho de Aquino et al. [8], os autores apresentam um algoritmo para redução baseada em *stream*

em RSSFs agrupadas, onde a redução dos dados gerados pelos nós integrantes do cluster é efetuada pelo nó líder. O objetivo nesse caso foi investigar se os algoritmos de redução baseada em *stream* de dados desenvolvidos para redes planas são eficientes também quando aplicados em uma rede hierárquica. Para isso, as técnicas de amostragem e rascunho, utilizadas para redução em redes planas em [6, 7], foram empregadas para a redução em redes hierárquicas. Nesse trabalho, é apresentada ainda uma descrição formal do problema de redução baseada em *stream* e um modelo analítico é adaptado para mostrar que as redes hierárquicas possuem desempenho superior às redes planas. Resultados mostraram que a técnica foi eficiente também em redes hierárquicas, conseguindo reduzir o atraso e o consumo de energia, com reduzidas perdas na qualidade dos dados.

Além dos trabalhos acima descritos, para o problema de redução de dados univariados, existem uma lista enorme de trabalhos que observam os dados objetivando identificar dados correlacionados e eliminar a redundância [3, 20, 31, 42, 84]. Também existem trabalhos que fazem fusão, compressão, correlação, redução ou agregação de dados, normalmente baseados na correlação das informações sensoriadas e com o objetivo de economizar recursos da rede, como energia, tempo de resposta e perda de pacotes [16, 34, 46, 59, 83, 84, 88, 89].

2.4.2 Redução de dados multivariados

Uma consideração importante é que grande parte dos trabalhos que utilizam técnicas de redução de dados em RSSFs não abordam a redução de dados multivariados, que são representados por conjuntos de valores de um ou mais nós, por exemplo, um nó que monitora temperatura, pressão e umidade simultaneamente ou um nó que processa os dados de um conjunto de nós que monitora apenas temperatura.

No trabalho de Seo et al. [73] é realizada uma comparação entre os métodos DWT (*Discrete Wavelet Transformation*), HCL (*Hierarchical Clustering*), Amostragem e SVD (*Singular Value Decomposition*) para redução de dados multivariados em RSSFs. Os métodos são avaliados variando o tamanho dos dados e o tipo de dado, utilizando conjuntos de dados reais e outros sintéticos. Segundo os autores, o método

de Amostragem teve desempenho superior com a variação do tamanho e do tipo de dados. Além disso, foi feita uma comparação entre o método SVD e o método de Redução Adaptativa, onde o dado é examinado primeiro e então escolhe o melhor método para efetuar a redução. Nesse caso, o SVD teve desempenho superior. O texto traz importantes contribuições à medida que introduz os possíveis caminhos e métodos para se aplicar a redução de dados multivariados em RSSFs e mostrando a importância de se considerar essas características. Entretanto, não fica claro no trabalho qual o melhor método para cada um dos tipos de aplicação considerados. Além disso, apesar do principal motivo de se aplicar a redução de dados em RSSFs ser o problema de energia, não foi feita uma avaliação do consumo de energia aplicando as técnicas citadas.

Também tratando da redução de dados multivariados em RSSFs, Cvejic et al. [22] apresentam um algoritmo para melhorar a fusão de imagens de vigilância, baseado na técnica Análise de Componentes Independentes (ICA), onde a codificação esparsa dos coeficientes de ICA diminuem o ruído nas imagens fundidas. O objetivo da fusão de imagens, além de diminuir o tráfego de informações, é criar imagens mais perceptíveis e adequadas para um processamento posterior. No método utilizado, para realizar a redução, um pré-processamento é realizado com a técnica de Análise de Componentes Principais (PCA), através da qual a dimensão dos dados é reduzida fazendo a decomposição dos autovalores da matriz de correlação dos dados, que indicam a significância dos vetores base. Posteriormente, os vetores base são estatisticamente selecionados, através de ICA. O próximo passo é a aplicação de um algoritmo para efetuar a fusão das imagens, em conjunto com um esquema para reduzir o ruído. Para reconstruir a imagem fundida, uma métrica de fusão é calculada em cada passo e quando a métrica de desempenho máximo da fusão é obtida, o processo pára e a imagem é reconstruída com os valores calculados. Resultados apresentados mostram a eficiência do método proposto em termos da qualidade das imagens após a fusão e também da redução de ruído. Um ponto importante desse trabalho é a utilização de PCA para realizar o pré-processamento, o que reforça a viabilidade de avaliá-la em

outros cenários e aplicações, além de mostrar a possibilidade de uso em conjunto com a técnica de fusão de dados.

Em [50], é apresentado um algoritmo que integra um método de compressão de dados baseado em PCA com o monitoramento das características do ambiente para realizar a predição e reduzir a quantidade de informações transmitidas. O contexto avaliado no trabalho é o monitoramento de dados sensoriados sobre vibração em larga escala para sistemas de monitoramento estrutural. No método proposto, as transmissões ocorrem progressivamente e com diferentes resoluções dos dados, dependendo do interesse. Resultados mostram que a técnica consegue diminuir a quantidade de dados com reduzidas perdas de informação, não citando, entretanto, valores para essas perdas. O foco do trabalho foi o monitoramento estrutural, mas os bons resultados encorajam a avaliação de PCA em outras aplicações e considerando outras distribuições de dados. Outro ponto que pode ser investigado é o uso de PCA em conjunto com técnicas de amostragem que não utilizam predição, uma vez que o seu mecanismo pode gerar um custo de comunicação maior na rede, conforme citado por Santini & Romer [71].

Devido à natureza de algumas aplicações, a integridade e a precisão dos dados são características fundamentais. Dessa forma, problemas causados por comprometimento ou mau funcionamento de alguns sensores precisam ser investigados. Em [18], é proposta uma abordagem para detecção de anomalias nos dados coletados por diferentes sensores em uma RSSF. Nessa proposta, PCA é utilizado com o propósito de reduzir a quantidade de dados, mantendo sua qualidade. Para efetuar essa redução, alguns componentes principais (número de componentes é variável e depende da necessidade) são utilizados para selecionar uma amostra com os dados que cobrem maior parte da variância do conjunto de dados originais. Após a seleção desses dados, uma teste estatístico para encontrar anomalias é empregado - o SPE (*squared prediction error*). Quando o limite previamente estabelecido para esse erro é excedido, indica que está ocorrendo uma anomalia, e é possível determinar os sensores responsáveis por essas informações anômalas. A avaliação do método é feita utilizando-se dados meteorológicos reais e inserindo-se anomalias nesses dados aleatoriamente, para então

medir sua capacidade de detecção. De acordo com os resultados apresentados, a técnica se mostrou eficiente, obtendo um bom nível de detecção. Fator interessante no trabalho é a utilização da matriz de correlação ao invés da matriz de covariância, mostrando que ambas podem ser empregadas e que podem obter resultados diferentes. É necessária, entretanto, uma avaliação mais detalhada a respeito da qualidade das amostras selecionadas através do PCA, com o objetivo de determinar sua qualidade em relação aos dados originais.

Conforme descrito por Roy & Vertterli [65], em diversas aplicações de RSSFs, os sensores enviam as informações coletadas para um centro de fusão, onde essas informações são fundidas para se recuperar o sinal original com a precisão desejada. Nesse trabalho, os autores propõem uma abordagem baseada em transformação para reduzir a quantidade de dados enviados para esse centro de fusão. Nessa técnica, cada sensor aplica uma transformação linear em seus dados através de PCA, para que seja enviada para o centro de fusão uma quantidade reduzida, a fim de minimizar o consumo de energia na rede. No centro de fusão, o sinal é reconstruído e uma análise da distorção desse sinal é efetuada através do erro médio quadrático (MSE - *mean squared error*). O trabalho tem como base aplicações nas quais as características de tempo e espaço são estacionárias, como por exemplo, a codificação de áudio. Os dados seguem uma distribuição gaussiana e utiliza o modelo de correlação Gauss-Markov de primeira ordem. Resultados mostram que a técnica obteve bons resultados, garantindo um nível de precisão adequado. Contudo, a avaliação é apenas teórica e com um foco diferente do abordado nessa dissertação, considerando aplicações e análises diferentes. No entanto, os bons resultados reforçam a viabilidade de investigação do uso de PCA em maiores detalhes e para outras aplicações.

Avaliando a técnica de Análise de Componentes Principais em outros cenários, Junior et al. [43] propõem o algoritmo OGK-multivar, que utiliza PCA em conjunto com a técnica de amostragem, abstraindo redundâncias e detalhes pouco significativos, transmitindo apenas as informações mais relevantes. A redução foi aplicada apenas no momento do sensoriamento, ou seja, no momento em que os nós coletam

os dados, a redução é realizada. O uso do método apresentado se mostrou bastante viável, em termos de consumo de energia e atraso inserido na rede. A técnica merece, entretanto, um estudo mais aprofundado no que se refere às distribuições utilizadas nos dados de entrada (apenas a distribuição gaussiana multivariada foi avaliada) e às métricas utilizadas para avaliar a qualidade dos dados após a redução, além da viabilidade de avaliação com diferentes topologias de rede. Outro cenário que pode ser considerado é a realização da redução durante o roteamento.

2.5 Exercícios de verificação de aprendizagem

1. Comente sobre as principais diferenças entre as transformadas de *Wavelets* que utilizam a base Haar, Daubechies e Coiflets.

2. Quais propriedades tornam a utilização da base de Coiflets mais atrativa para aplicações de redução de dados?

3. Quais as principais diferenças entre componentes principais e componentes independentes?

4. O teste KS é um teste que independe da normalidade dos dados. Com isso, ele pode apresentar resultados pessimistas em conjuntos de dados específicos. Faça uma busca a respeito de outros testes que possam ser utilizados para tipos específicos de dados e que apresentem resultados mais satisfatórios quando comparados com o teste KS.

5. Considerando a análise de dados multivariados, apresente testes alternativos ao teste ANOVA que podem ser utilizados em conjunto de dados genéricos ou específicos.

Capítulo 3

Arcabouço para redução de dados

Neste capítulo, apresentaremos uma visão geral de um arcabouço para redução de dados, bem como o detalhamento dos elementos que o compõe. Tais elementos, consistem na caracterização, suporte a redução, robustez e concepção.

3.1 Visão geral do arcabouço

Apresentamos aqui um arcabouço para redução de dados em RSSFs que assume como premissa a necessidade emergente na "padronização" das técnicas e ferramentas utilizadas nas aplicações de RSSFs. Nessa direção, a principal hipótese considerada é que:

> *A utilização de um arcabouço permite uma prototipagem eficaz de novas aplicações em RSSFs. Tal prototipagem considera as fases de projeto, análise, implementação, implantação e teste das aplicações. Vale salientar que as aplicações aqui mencionadas envolvem todos os aspectos de hardware, funções de rede e software.*

Nesse arcabouço são previstas quatro etapas para o desenvolvimento de uma aplicação em RSSFs: caracterização, suporte à redução,

robustez e concepção. Estas etapas estão ilustradas na figura 3.1 e serão detalhadas mais adiante.

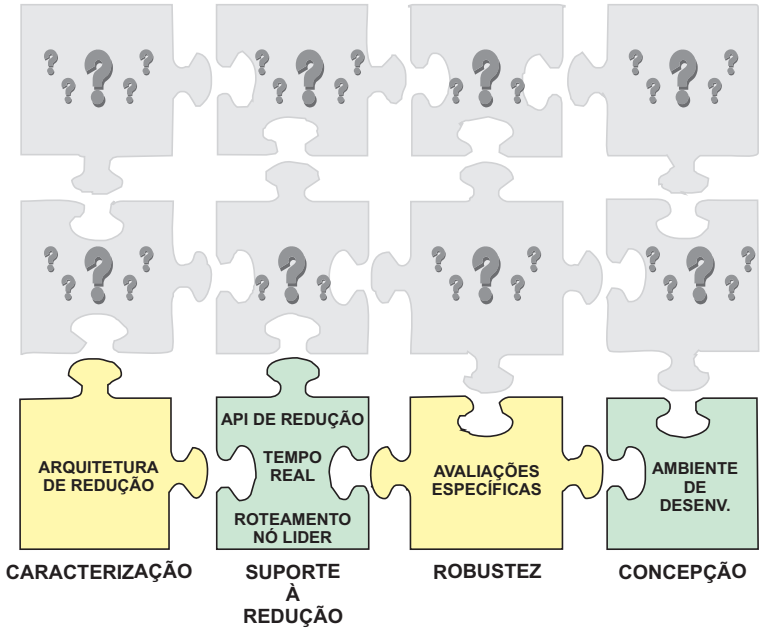


Figura 3.1: Etapas para o desenvolvimento de aplicações de redução de dados.

Considerando as necessidades dos projetistas, esse arcabouço, provê:

- A definição de um modelo conceitual para o projeto das aplicações de redução;
- A implementação de uma estrutura de suporte para viabilizar a utilização de soluções de redução, tanto em *software* como em *hardware*;
- A definição de um modelo conceitual para a avaliação da qualidade das soluções de redução;
- A disponibilização de um ambiente para teste, desenvolvimento e implantação, das soluções presentes no arcabouço, em ambientes reais.

Como esperado, para cada problema relacionado à essas etapas encontramos facilmente soluções específicas para muitos cenários e aplicações na literatura. Isso evidencia uma carência na padronização e/ou na existência de ambientes com diferentes soluções implementadas e integradas que facilitem o desenvolvimento, implantação e a comparação de novas técnicas com as existentes. Essa especificidade é a grande motivação para a utilização deste arcabouço.

As demais soluções encontradas na literatura (representadas pelas interrogações) podem ser: independentes, se encaixarem sequencialmente na forma horizontal ou vertical; ou sobrepostas. O que buscamos no arcabouço é o melhor encaixe dessas peças a fim de favorecer o desenvolvimento rápido e padronizado das aplicações de redução de dados em RSSFs. A seguir apresentamos o detalhamento da figura 3.1, que representa a base do arcabouço.

- **Arquitetura de redução:** Aquino [4] propõe uma arquitetura para o desenvolvimento de aplicações de redução baseada em *stream* de dados em RSSFs. Partindo dessa arquitetura é possível descrever um modelo conceitual para redução de dados, genérico o suficiente, que pode ser utilizado para diferentes tipos de aplicações. Tal generalização considera: a caracterização dos dados sensoriados, ou seja, como os dados podem ser coletados e representados; e a caracterização das aplicações, ou seja, quais os requisitos inerentes às aplicações como, por exemplo, qual o prazo para entrega de dados, quanta energia pode ser gasta para obter um conjunto de leituras do ambiente ou qualquer outro parâmetro de QoS (qualidade de serviço).

- **API de redução:** A arquitetura proposta disponibiliza um conjunto de algoritmos de redução baseado em *stream* de dados. Tais algoritmos consideram técnicas de amostragem e rascunho que podem ser aplicadas para dados univariados e multivariados [6, 7, 14, 43, 44]. Outros algoritmos podem facilmente compor essa API, tanto algoritmos para redução de dados como algoritmos a serem utilizados em conjunto com as funções de rede como, por exemplo, roteamento ou controle de densidade. Além disso, os algoritmos podem ser disponibilizados para as aplicações como,

por exemplo, aplicações de consulta, gerenciamento da rede ou reconfiguração de *software*.

- **Avaliações específicas:** Para todos os cenários considerados em [6, 43], avaliações específicas para a qualidade dos dados foram realizadas. Tais avaliações são necessárias, pois uma vez que reduzimos os dados precisamos identificar se essa redução mantém a semântica dos dados originais. Para tal, técnicas estatísticas para identificação da fidelidade dos dados univariados e multivariados foram utilizadas. No entanto, para identificar robustez da aplicação de redução utilizada, devemos considerar também outros aspectos, por exemplo: identificar, por intermédio de modelos matemáticos, qual o ganho real da rede ao se utilizar redução de dados; determinar a metodologia de avaliação que deve ser considerada para cada tipo de dado ou aplicação identificada na etapa de caracterização.

- **Ambiente de desenvolvimento:** Aquino et al. [12, 13] propõem uma ferramenta para o desenvolvimento de aplicações de monitoramento de ambientes móveis. Com essa ferramenta é possível a prototipagem de aplicações de redução de dados que podem ser testadas e utilizadas em sensores reais. De forma complementar, generalizações e melhoramentos das ferramentas de simulação são necessários para acelerar esse processo de prototipagem. Uma vez que o protótipo esteja funcional, é necessário a definição de uma metodologia para se efetuar a implantação das aplicações em cenários reais. Tal metodologia envolve aspectos de caracterização de interface de comunicação, características de *hardware*, entre outros aspectos relacionados à concepção das aplicações.

Partindo do conhecimento já obtido e resumido acima, o arcabouço conceitual para redução de dados em RSSFs, com todas as suas subetapas, é ilustrado na figura 3.2.

Uma breve descrição de cada subetapa da figura 3.2 é apresentada a seguir:

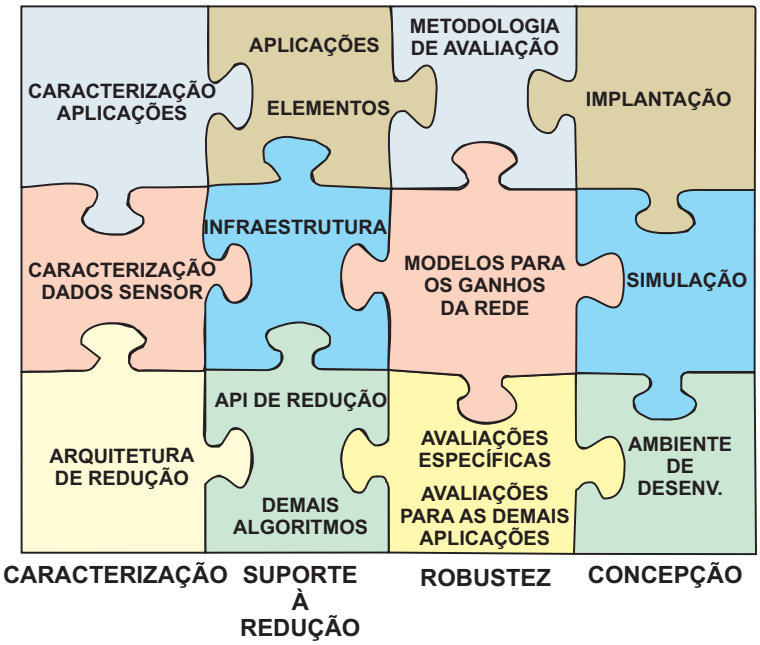


Figura 3.2: Arcabouço conceitual para reduções de dados em RSSFs.

- **Caracterização:** Essa etapa prevê a caracterização e o levantamento de requisitos das aplicações de redução em RSSFs. A caracterização contempla as seguintes subetapas:
  - *Arquitetura de redução*, que pode ser constantemente refinada e instanciada para cada cenário considerado;
  - *Caracterização dos dados sensoriados* para determinar como os modelos que descrevem os dados devem ser definidos; esses modelos são importantes para identificar o comportamento dos dados, permitindo que algoritmos específicos sejam propostos;
  - *Caracterização das aplicações* para permitir a catalogação dos requisitos das aplicações considerados no projeto dos algoritmos de redução, por exemplo, a definição dos requisitos de QoS que interferem nos algoritmos.

- **Suporte à redução:** Essa etapa prevê o suporte à implementação para as aplicações de redução. O suporte à redução contempla as seguintes subetapas:
  - *API de redução*, que possui todos os algoritmos de redução disponíveis para serem utilizados em conjunto com as funções de redes ou nas aplicações;
  - *Funções de rede* que representam os mecanismos de rede que utilizam internamente a redução de dados, por exemplo, roteamento, controle de densidade ou criação de agrupamentos baseada nos dados sensoriados;
  - *Aplicações* que necessitam de mecanismos de redução explícitos, por exemplo, consultas de usuário à rede – nesse caso os nós sensores precisam armazenar um rascunho dos dados (i.e., dados reduzidos) e reconstruí-los para responder às consultas efetuadas.
- **Robustez:** Essa etapa prevê a definição de modelos que permitam a validação e quantificação dos algoritmos e mecanismos incorporados ao arcabouço. A robustez contempla as seguintes subetapas:
  - *Avaliações específicas* que são necessárias para os casos onde modelos genéricos para a representação dos dados sensoriados são factíveis de serem definidos; logo, avaliações específicas podem ser aplicadas, obtendo-se um melhor refinamento na classificação entre os algoritmos propostos;
  - *Modelos para os ganhos da rede* que são necessários quando os requisitos da rede devem ser minimizados ou quando a satisfação do usuário da rede deve ser maximizada; nesse caso diferentes problemas de otimização podem ser investigados para permitir que os objetivos dos projetistas da rede sejam alcançados;
  - *Metodologia de avaliação* que prevê a definição de modelos conceituais que incorporem a avaliação em conjunto dos algoritmos de redução com os modelos de otimização conside-

rando os aspectos de algoritmos de redução, funções de rede e aplicações.

- **Concepção:** Essa etapa prevê o incremento dos mecanismos de simulação existentes, permitindo a validação dos mecanismos propostos. Além disso, por intermédio de estudos de caso, esta etapa prevê a caracterização de ambientes reais das aplicações concebidas no âmbito do projeto. A concepção contempla as seguintes subetapas:
  - *Ambiente de desenvolvimento* gráfico com todos os algoritmos disponíveis permitindo uma fácil e rápida prototipagem das aplicações;
  - *Simulação* que permite incorporar aos simuladores dados caracterizados, algoritmos e funções de rede voltados à redução de dados e avaliações específicas para o conjunto de dados disponível no simulador;
  - *Implantação* das soluções de redução de dados em ambientes reais, de tal forma que os resultados dos experimentos possam ser utilizados para realimentar e melhorar todo o arcabouço conceitual.

De forma geral, o arcabouço conceitual torna-se uma boa fonte de estudos pois boa parte dos trabalhos encontrados na literatura não consideram a relação "dados vs. funções de redes" nem avaliam satisfatoriamente a qualidade da redução.

3.2 Caracterização

As aplicações em RSSFs possuem diferentes configurações, tipos de dados e requisitos. Assim, é importante termos um modelo conceitual que guie ou direcione os projetistas das redes, para os casos onde é necessário efetuar reduções de dados. O modelo conceitual será o principal extrato da etapa de caracterização. Algumas abordagens consideradas às subetapas da caracterização são listadas abaixo:

- **Caracterização dos dados sensoriados:** Para o processamento dos dados de sensoriamento é importante conhecer o comportamento do fenômeno monitorado e a forma com que a aplicação obtém as amostras do ambiente. Com isso, temos:

  - *Problema abordado*: Uma vez que estamos interessados em discretizar algum fenômeno, como podemos caracterizar esse fenômeno e representá-lo no domínio espaço temporal?
  - *Hipótese 1*: Podemos modelar o fenômeno por intermédio de modelos estatísticos.
  - *Hipótese 2*: Com os modelos estatísticos que descrevem os dados monitorados, podemos propor algoritmos mais eficientes e específicos para cada tipo de fenômeno.
  - *Hipótese 3*: É possível generalizar as estratégias de redução de dados com base em modelos com comportamentos similares.
  - *Conhecimento prévio*: Frery et al. [29] propõem uma representação para o campo de sensoriamento que é utilizada para caracterizar apenas um instante de todo o espaço monitorado.

- **Caracterização das aplicações:** Além da exigência de monitorar os dados com eficácia as aplicações possuem requisitos que, muitas vezes, devem ser cumpridos (consumo de energia, atraso etc). Com isso, temos:

  - *Problema abordado*: Uma vez que existem requisitos associados às aplicações em RSSFs, como identificar e avaliar requisitos que podem ser atendidos por intermédio das soluções de redução?
  - *Hipótese 1*: Por intermédio da catalogação dos requisitos associados às aplicações, podemos identificar "como e onde" utilizar os algoritmos de redução de dados em conjunto como as funções de rede.

  - *Hipótese 2*: Podemos propor modelos matemáticos que descrevem o impacto da redução sobre cada um dos requisitos da aplicação.
  - *Conhecimento prévio*: Aquino et al. [10, 11] apresentam uma catalogação dos requisitos das aplicações de tempo real em RSSFs, propõem um modelo matemático para identificar a quantidade de dados que podem ser propagados até o sorvedouro e apresentam um algoritmo de roteamento sensível aos dados que garante a entrega dos dados no prazo estabelecido pela aplicação.

3.3 Suporte à redução

Ao utilizarmos o arcabouço para o suporte e desenvolvimento de aplicações de redução é necessário considerar: os algoritmos de redução, a integração das técnicas de redução de dados às funções da rede e a utilização da redução em aplicações específicas. Vale destacar que a maior parte dessas soluções de suporte à redução podem ser implementadas em *software* ou em *hardware*. Algumas abordagens consideradas às subetapas do suporte à redução são:

- **Algoritmos de redução:** Além dos algoritmos de redução de dados já disponíveis, outros estudos precisam ser efetuados em relação aos algoritmos de redução de dados em geral. Analisando pontualmente a redução de dados, temos:
  - *Problema abordado*: Uma vez que não é viável o envio de todos os dados monitorados, é possível efetuar a redução desses dados mantendo a representatividade dos dados e economizando recursos da rede?
  - *Hipótese 1*: Ao utilizarmos algoritmos de redução de dados é possível economizar recursos da rede mantendo a representatividade dos dados.
  - *Hipótese 2*: Se considerarmos uma rede com um alto tráfego, a perda de pacotes pode degradar os dados enviados

por completo e fazer com que os dados reduzidos possuam uma representatividade melhor que os dados completos enviados.

  - *Hipótese 2*: Ao considerarmos, as aplicações de consultas sobre as RSSFs, podemos utilizar apenas o rascunho dos dados armazenados respondendo essas consultas de forma aproximada. A saber, a principal razão para utilizarmos apenas o rascunho sobre os dados e não todos os dados previamente monitorados é o alto custo de armazenamento associado ao nó sensor.
  - *Conhecimento prévio*: Aquino et al. [6, 7, 14, 43, 44] apresentam uma série de algoritmos de redução baseada em técnicas de *stream* de dados e mostram que a hipótese acima mencionada pode ser alcançada para essa classe de algoritmos. Dentre esses algoritmos, o que possui resultados iniciais e ainda está em fase de aprimoramento é o algoritmo que considera técnicas de *stream* de dados baseada em *wavelets*. Esse algoritmo permite a detecção dinâmica de eventos ou ruídos presentes nos dados monitorados [14].

- **Integração da redução de dados às funções de rede:** As funções de rede podem se beneficiar das informações contidas nos dados ou em um simples rascunho e tomar decisões locais mais adequadas à aplicação. Com isso, temos:

  - *Problema*: É possível utilizar a redução de dados para auxiliar as funções de rede?
  - *Hipótese 1*: Por definição, as RSSFs são redes centradas nos dados. Com isso, ao disponibilizarmos as soluções de controle de densidade informações de sensoriamento (com o rascunho dos dados) podemos efetuar o desligamento apenas dos nós que monitoram dados redundantes permitindo que apenas as informações importantes sejam enviadas. A saber, o problema do controle de densidade objetiva utilizar o menor número de nós na rede, de tal forma que a cobertura e

conectividade seja mantida, como apresentado por Menezes et al. [57].

  - *Hipótese*: O roteamento pode tratar informações a respeito dos requisitos da rede, como nível de energia, atraso dos pacotes e qualidade dos dados. Podemos durante o roteamento efetuar reduções de dados para atender esses requisitos.
  - *Conhecimento prévio*: O trabalho proposto por Aquino & Nakamura [11] apresenta um algoritmo de roteamento sensível aos dados para aplicações de tempo real. Além disso, outros trabalhos relacionados a funções de redes sem considerar redução de dados foram realizados: auto-organização [27], reconfiguração da rede [54], roteamento [61] e caracterização topológica da rede [33, 67, 68, 69].

- **Redução em aplicações específicas:** De forma complementar à utilização da redução de dados em conjunto com as funções de redes, podemos considerar soluções de redução diretamente para aplicações específicas. Diferente dos tópicos anteriormente discutidos, ao invés de apresentarmos o problema a ser abordado, discutiremos apenas as implicações da redução de dados nas aplicações. Com isso, temos:

  - *Qualidade de serviço*: Considerando que as aplicações em RSSFs podem exigir qualidade de serviço, podemos considerar a qualidade dos dados como um parâmetro de QoS, ou seja, a redução de dados irá ocorrer de tal forma que o nível de qualidade seja sempre mantido. Outro aspecto, é a identificação de parâmetros de rede que podem ser degradados pelo volume de dados, por exemplo, energia, atraso e perda de pacotes. É possível ajustar a redução de dados para que esses parâmetros sejam atendidos.
  - *Tempo real*: Em RSSFs, geralmente, consideramos as aplicações de tempo real *soft* pois, o ambiente não é controlado e as aplicações utilizam métodos aproximados para tratar os dados e atender aos prazos exigidos. Assim, podemos utilizar junto ao roteamento a redução de dados com base no prazo

das aplicações. Nesse caso, o conhecimento já obtido pode ser destacado pelos trabalhos desenvolvidos por Aquino et al. [5, 9, 10, 11].

– *Abstração da RSSF como um banco de dados*: Existem aplicações que necessitam efetuar consultas às informações geradas pelos sensores. Essas informações podem ser referentes aos dados antigos, por exemplo, quais as regiões monitoradas que tiveram em algum momento temperaturas maior que 50 graus. Nessa direção, é importante termos algoritmos de rascunho de dados, com uma aproximação aceitável, para um conjunto de aplicações. Além disso, é necessário o armazenamento econômico das informações nos nós sensores que facilite as consultas feitas aos nós.

3.4 Robustez

Ao reduzirmos os dados é necessário avaliar se sua representatividade é aceitável para a aplicação. Além disso, por estarmos considerando melhorias nas métricas de rede, modelos matemáticos são necessários para garantir que essas melhorias sejam próximas das ótimas. Algumas abordagens consideradas às subetapas da robustez são listadas abaixo:

- **Metodologia para a avaliação da qualidade dos dados:** Uma vez que estamos efetuando a redução de dados é importante que, para cada tipo de dado de entrada e para cada tipo de técnica utilizada, façamos uma avaliação do erro em relação aos dados originais. Com isso, temos:

  – *Problema*: Qual a melhor abordagem para se efetuar a avaliação da qualidade dos dados, específica ou genérica?

  – *Hipótese*: Assumindo que temos a caracterização dos dados a melhor estratégia é a específica. Caso contrário estratégias generalizadas são mais indicadas.

  - *Conhecimento prévio*: Apenas estratégias generalizadas foram consideradas em trabalhos anteriores. O trabalho proposto por Aquino et al. [6] apresenta estratégias baseadas em métodos estatísticos que avaliam a representatividade dos dados seguindo a sua distribuição e os valores no conjunto de dados amostrado. O trabalho de Frery et al. [29] apresenta uma avaliação baseada em reconstrução da área sensoriada.

- **Modelos matemáticos que otimizem os parâmetros de rede:** Na maior parte das aplicações de redução de dados em RSSFs, estamos interessados em obter ganhos em relação aos parâmetros da rede e manter a qualidade dos dados. É importante, para cada parâmetro da rede determinar, matematicamente, qual a melhor configuração de redução considerando cenários específicos. Com isso, temos:

  - *Problema*: Como garantir a qualidade dos dados reduzidos e a economia dos parâmetros da rede em um limiar próximo ao ótimo?
  - *Hipótese*: Com a definição de modelos de otimização mono ou multiobjetivo e da utilização de heurísticas distribuídas podemos garantir a qualidade dos dados e a economia dos parâmetros da rede.
  - *Conhecimento prévio*: Vários trabalhos foram desenvolvidos considerando problemas de otimização em RSSFs [33, 57, 67, 68, 69]. Estes trabalhos consideram problemas de controle de densidade e caracterização topológica das RSSFs. Ademais, são utilizados métodos exatos, como a decomposição de benders, e heurísticas como os algoritmos evolutivos.

3.5 Concepção

Por fim, ferramentas de simulação e desenvolvimento que permitam a implantação das soluções presentes no arcabouço em ambientes reais são integradas ao arcabouço. Como essa etapa está mais direcionada às ferramentas apenas discutiremos o que precisa ser feito no

arcabouço para contemplar a concepção das aplicações de redução de dados em RSSFs. Algumas abordagens consideradas às subetapas da concepção são listadas abaixo:

- **Simulação:** Diversas soluções para RSSFs são avaliadas por intermédio de simulação utilizando algumas ferramentas, como o *Network Simulator*[1] e Sinalgo[2]. É previsto a integração das soluções de redução disponíveis no arcabouço, nesses simuladores, para permitir que novas soluções de sejam facilmente testadas e simuladas. Nessa direção, podemos incluir, por exemplo, geradores de diferentes tipos de dados de sensoriamento, geração de tráfego presente nas aplicações de redução e sumarização de resultados de simulação direcionados a parâmetros específicos. Além disso, considerando apenas os algoritmos de redução, é importante disponibilizarmos um ambiente para a avaliação da qualidade dos dados separadamente, ou seja, uma ferramenta com os recursos necessários para a avaliação e validação dos algoritmos.

- **Implantação:** Considerando nós sensores que utilizam como base o sistema operacional TinyOS e a linguagem NesC[3], é possível integrar ao arcabouço, uma infraestrutura básica, baseada em NesC, que possa ser reutilizada em diferentes aplicações de redução. Assim, tanto as soluções existentes como as novas, podem ser testadas em uma RSSF real. De forma complementar, considerando o conjunto de algoritmos e soluções presentes no arcabouço, é possível, adaptar a ferramenta de desenvolvimento baseada em componentes, proposta por Aquino et al. [12, 13], para permitir a geração de aplicações de redução em NesC. Dessa forma, será possível desenvolver aplicações reais, baseada em componentes, em um tempo reduzido.

[1] `http://nsnam.isi.edu/nsnam/index.php/Main_Page`
[2] `http://dcg.ethz.ch/projects/sinalgo/`
[3] `http://www.tinyos.net/`

3.6 Exercícios de verificação de aprendizagem

1. Quais as principais etapas de desenvolvimento previstas pelo arcabouço de redução? Liste e apresente suas principais características.

2. Considerando as avaliações específicas, apresente alguns exemplos de técnicas que podem ser utilizadas para os cenários de redução de dados.

3. Considerando uma aplicação para detecção de incêndios, apresente uma caracterização dos dados sensoriados para essa aplicação.

4. Apresente uma lista de problemas que podem ser abordados na subetapa de *Modelos para os ganhos da rede*.

5. Faça uma breve busca e liste algumas ferramentas que podem ser utilizadas como ambientes de desenvolvimento para aplicações de redução de dados.

6. Para a subetapa de *Redução em aplicações específicas*, foram listadas três aplicações que podem ser utilizadas: QoS, tempo real e banco de dados. Apresente uma outra aplicação que possa utilizar a redução de dados em seu benefício. Detalhe a aplicação e qual seria o papel da redução de dados em tal aplicação.

Capítulo 4

Algoritmos de redução de dados

Apresentamos aqui um conjunto de algoritmos que podem ser utilizados em conjunto, ou como parte, do arcabouço para redução de dados em RSSFs, apresentado no capítulo anterior. Consideraremos algoritmos de amostragem para dados univariados e multivariados. Selecionamos os seguintes algoritmos:

- **Rascunho de dados**: Esse algoritmo [6] considera a montagem do histograma dos dados e armazena apenas um rascunho do histograma, ou seja, armazena as informações de quantidade de elementos, elementos de menor e maior valores, a quantidade de classes e a porcentagem de elementos em cada classe.
- **Amostragem aleatória**: Esse algoritmo [7] considera a montagem do histograma dos dados e efetua a amostragem de forma aleatória sob cada classe do histograma. Com isso, a distribuição dos dados é preservada.
- **Amostragem dos dados centrais**: Esse algoritmo [11] é similar ao algoritmo aleatório, porém efetua a amostragem considerando os dados mais centrais sob cada classe do histograma.
- **Amostragem Wavelets**: Esse algoritmo [14] considera a amostragem baseada nos coeficientes de *Wavelets* obtidos ao utilizar

a base de Coiflets.

- **Amostragem baseada em componentes principais**: Esse algoritmo [43, 44] é destinado para dados multivariados e considera uma classificação prévia dos dados utilizando análise de componentes principais ou independentes.

4.1 Rascunho de dados

De forma geral, a técnica de rascunho utiliza informações dos dados de entrada, como mínimo, máximo, média e frequência para inferir propriedades dos dados. A técnica de rascunho mantém a frequência dos dados sem perda e utiliza um tamanho fixo de pacote para o envio da informação sensoriada, o que permite economizar recursos da rede. O objetivo, em utilizar essa técnica, é obter a distribuição de frequência dos dados de tal forma que o dado original possa ser gerado fora da rede. Considerando V o dado de entrada, o algoritmo rascunho segue os seguintes passos (figura 4.1):

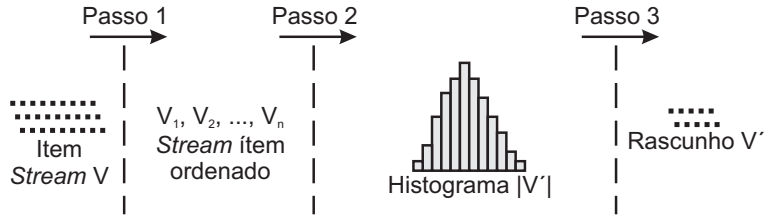


Figura 4.1: Passos utilizados para o processamento do dado de entrada no algoritmo rascunho.

Passo 1 Ordenação dos dados e identificação dos valores mínimo e máximo dentro do dado de entrada V.

Passo 2 Construção dos dados de saída apenas com as freqüências do histograma.

Passo 3 Montagem do rascunho V' com os dados de saída obtidos no passo 2 e as informações a respeito do histograma obtidas no passo 1.

O pseudo-código do algoritmo rascunho pode ser visto no algoritmo 1 e sua análise de complexidade é apresentada a seguir:

Algorithm 1: Pseudo-código do algoritmo de rascunho.

```
   Entrada: V – item de entrada original
   Saída: V' – rascunho resultante
 1 início
 2 |  Ordene(V);
 3 |  lg ← "Largura de cada classe do histograma";
 4 |  |V'| ← ⌈(V[|V|] − V[0])/lg⌉;
 5 |  pr ← V[0] {primeiro elemento da classe do histograma};
 6 |  c ← 0 {contador};
 7 |  indice ← 0;
 8 |  para i ← 0 até |V| − 1 faça
 9 |  |  se V[i] > pr + lg ou i = |V| − 1 então
10 |  |  |  v'[indice] ← c;
11 |  |  |  indice ← indice + 1;
12 |  |  |  c ← 0;
13 |  |  |  pr ← V[i];
14 |  |  fim
15 |  |  c ← c + 1;
16 |  fim
17 fim
```

Linha 2 Executa em $O(|V| \log |V|)$.

Linhas 3–7 Correspondem à inicialização das variáveis.

Linhas 8–16 Definem o laço para construção do histograma, executado em $O(|V|)$.

Assim, a complexidade do algoritmo rascunho é

$$O(|V| \log |V|) + O(|V|) = O(|V| \log |V|).$$

A complexidade de espaço é $O(|V| + |V'|) = O(|V|)$, pois armazenamos do dado de entrada V e a amostra V' resultante. Já que todo nó envia sua amostra em direção ao sorvedouro, a complexidade de comunicação é $O(|V'| D)$, onde D é a maior rota (em saltos) da rede.

4.2 Amostragem aleatória e central

De forma geral, a técnica de amostragem processa cada dado de entrada, aplicando algum tipo de seleção dos dados mais significativos. A idéia principal é obter amostras suficientes para podermos representar o fenômeno monitorado. O objetivo, em utilizar essa técnica, é maximizar a relação qualidade dos dados vs. manutenção da rede. Na seguinte implementação, o tamanho do conjunto de amostras pode ser regulado de forma *online*.

O algoritmo de amostragem possui duas variantes: *aleatório* que faz a escolha das amostras de forma aleatória; e *central* que escolhe os elementos centrais das colunas do histograma gerado a partir dos dados originais. Considerando V o dado de entrada, o algoritmo de amostragem segue os seguintes passos (Figura 4.2):

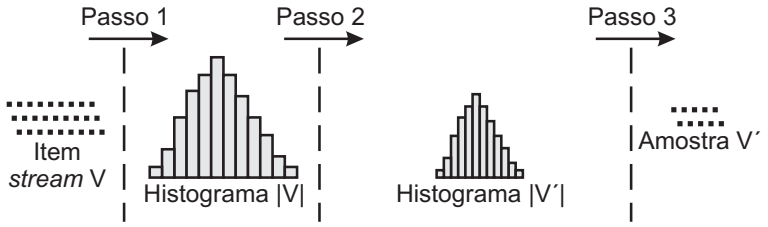


Figura 4.2: Passos utilizados para o processamento do dado de entrada no algoritmo de amostragem.

Passo 1 Construção do histograma a partir do item original V. No momento da construção do histograma o número de classes utilizadas interfere na qualidade do resultado.

Passo 2 (*aleatório*) Criação de um novo histograma com o tamanho $|V'|$, a partir do histograma obtido no passo 1. Esse novo histograma possui a mesma distribuição de frequência do original e os valores que irão compor suas colunas são escolhidos aleatoriamente no histograma original. Esses valores são utilizados para compor V'.

Passo 2 (*central*) Criação de um novo histograma com o tamanho $|V'|$, a partir do histograma obtido no passo 1. Para criar esse

novo histograma, foram escolhidos os elementos centrais de cada classe do histograma original. A amostra resultante V' será representada pelo novo histograma.

Passo 3 Ordenação de V' de acordo com a ordem de chegada de V.

O pseudo-código do algoritmo de amostragem pode ser visto no algoritmo 2.

Algorithm 2: Pseudo-código do algoritmo de amostragem.

```
   Entrada: V – item de entrada original
   Entrada: |V'| – tamanho da amostra resultante
   Saída: V' – amostra resultante
 1 início
 2     Ordene(V);
 3     lg ← "Largura de cada classe do histograma";
 4     pr ← 0 {primeiro índice da primeira classe do histograma};
 5     n_c ← 0 {número de elementos, por classe do histograma};
 6     k ← 0;
 7     para i ← 0 até |V| − 1 faça
 8         se V[i] > V[pr] + lg ou i = |V| − 1 então
 9             n'_c ← ⌈n_c |V'|/|V|⌉ {número de elementos por coluna em V'};
10             indice ← "Escolha do índice seguindo o passo 2";
11             para j ← 0 até n'_c faça
12                 V'[k] ← V[indice];
13                 k ← k + 1;
14                 indice ← "Escolha do índice seguindo o passo 2";
15             fim
16             n_c ← 0;
17             pr ← i;
18         fim
19         n_c ← n_c + 1;
20     fim
21     Ordene(V') {de acordo com a ordem original};
22 fim
```

No algoritmo 2 temos duas possibilidades para a execução das linhas 10 e 14, que representam a escolha das amostras que irão compor o item V' de saída. No caso da escolha aleatória, temos para ambas

as linhas

$$indice \leftarrow Aleatoria(pr, pr + n'_c),$$

onde a função *Aleatoria*, retorna algum número inteiro entre $[pr, pr + n'_c]$. Já no caso da escolha pelos elementos centrais da coluna do histograma, temos na linha 10

$$indice \leftarrow pr + \lceil (n_c - n'_c)/2 \rceil$$

e na linha 14 temos $indice \leftarrow indice + 1$. Fazendo a análise de complexidade do algoritmo 2 temos:

Linha 2 Executa em $O(|V| \log |V|)$;

Linhas 3–6 Correspondem à inicialização das variáveis.

Linhas 11–15 Definem o laço interno que determina o número de elementos de cada classe do histograma da amostra resultante. Considere, H_{cn} o número de classes dos histogramas. O tempo de execução do laço interno é de $O(|V'|)$, onde na linha 11 $n'_c = |V'| \leftrightarrow H_{cn} = 1$, ou seja, teríamos uma única classe no histograma da amostra com $|V'|$ elementos a serem percorridos.

Linhas 7–20 Definem o laço externo onde a entrada dos dados é lida e os elementos da amostra são escolhidos. H_{cn} é o número de classes dos histogramas. Antes da linha 8 ser aceita, executamos n_c vezes o laço externo, o que corresponde a contagem do número de elementos de uma classe do histograma original (linha 19). Após a condição da linha 8 ser aceita, o laço mais interno é executado n'_c vezes. A condição da linha 8 é aceita apenas H_{cn} vezes. Com isso, temos uma execução de $H_{cn}\,(n_c + n'_c)$ para o laço mais externo. Como $|V| = H_{cn}\, n_c$ e $|V'| = H_{cn}\, n'_c$, temos um tempo de execução para o laço externo de $O(|V| + |V'|)$.

Linha 21 Executa em $O(|V'| \log |V'|)$.

Assim, a complexidade do algoritmo de amostragem é

$$O(|V| \log |V|) + O(|V| + |V'|) + O(|V'| \log |V'|) = O(|V| \log |V|),$$

já que $|V'| \leq |V|$. A complexidade de espaço é $O(|V| + |V'|) = O(|V|)$ pois armazenamos o dado de entrada V e a amostra V' resultante. Já que todo nó envia sua amostra em direção ao sorvedouro, a complexidade de comunicação é $O(|V'|\, D)$, onde D é a maior rota (em saltos) da rede.

4.3 Amostragem Wavelets

Algoritmo de amostragem baseado em transformada de *Wavelet* aqui apresentado utiliza a transformada *Wavelet* com funções de base de Coiflets, permitindo assim uma amostragem dinâmica dos dados. Por intermédio desse tipo de amostragem, é possível uma detecção mais detalhada de eventos, comparando com outras funções de base da *Wavelet*, como por exemplo, Haar ou Daubechies. Isso ocorre, pois a base Coiflets possui melhores resultados quando os dados podem ser interpolados por uma função polinomial. Com isso, o algoritmo de Wavelet, ao ser aplicado em RSSFs, reduz os dados com eficiência, sem perder a representatividade dos mesmos.

Seja V'_j uma sequência de subespaços fechados de $L^2(\mathbb{R})$ e $f(x) \in L^2(\mathbb{R})$ seja o sinal observado. Cada V'_j representa aproximações sucessivas do sinal original, considerando a resolução de 2^j. Os detalhes da projeção em 2^j e 2^{j-1}, denotado por W_j, é definido por $W_j \bigoplus V'_j = V'_{j-1}$, onde $\bigoplus$ denota a soma direta de dois espaços vetoriais.

Filtros discretos são definidos para escolher os níveis de frequência presentes nos dados, que variam em escala temporal. Dois conjuntos de funções são aplicados: funções de escala $\phi(t)$ e funções de *Wavelet* $\psi(t)$. Dessa forma, pode ser aproximado através da seguinte expansão:

$$f(x) = \sum_n s_{i_0}[n]\, \psi_{i_0,n(x)} + \sum_{i=i_0}^{i_1} w_i[n]\, \phi_{i_0,n(x)} \tag{4.3.1}$$

onde $s_{i_0}[n] \in V'_j$ são coeficientes de escala e $w_i[n] \in W_i$ são coeficientes de *Wavelet*

O algoritmo baseado em *Wavelets* pode ser dividido em alguns passos:

Passo 1 Definir V como o dado sensoriado e gerar as funções $h(i)$ e $g(i)$ da base Coiflets. Esses filtros h e g são a forma discreta de dois tipos de funções aplicadas à transformada Wavelet, funções de Wavelet $\psi(t)$ e funções de escala $\phi(t)$.

Passo 2 Aplicar a transformada de *Wavelet* por intermédio de g e h. O resultado será a decomposição do sinal em diferentes subespaços, cada um com diferentes resoluções de tempo e frequência.

Passo 3 Calcular uma taxa aproximada de erro para manter a representatividade dos dados, incluindo quando algum evento externo ocorrer. O erro aproximado é:

$$\left\| f(x) - \sum_{\tau} g(\tau)\, \phi_{s,\tau(x)} \right\| = O(2^{s.2L}) \tag{4.3.2}$$

onde as variáveis s e τ são as novas dimensões obtidas depois da transformada, escala e translação; e $2L$ é o número de momentos Coiflets.

O algoritmo 3 apresenta o pseudocódigo da redução baseada em transformada *wavelet*.

Seja M o número de níveis decompostos, então o número total de operações no vetor de tamanho $|V|$ terá o número de operações na ordem de $O(M\,|V|\,\log|V|)$ [14].

4.4 Amostragem baseada em componentes

Para ilustrar os dados multivariados gerados nas aplicações, considere uma matriz V_n^s os dados de entrada, onde $n > 0$ representa os valores monitorados por cada sensor e $s \geq 1$ representa os sensores responsáveis por obter informações do ambiente. Com isso, para descrever o funcionamento básico do algoritmo para aplicar, é possível considerar os seguintes passos, ilustrados na figura 4.3.

Algorithm 3: Algoritmo baseado em transformada *wavelet*

Entrada: V – dados originais, h, g – filtros da base Coiflets
Entrada: V'_j – amostra resultante
Saída: Filtros $h(i)$ e $g(i)$

```
1  início
2      para t ∈ [0, |V|/2 − 1] faça
3          u = 2t + 1;
4          V'_jt = g_1 V_{u+1};
5          para n ∈ [1, |h| − 1] faça
6              u = u − 1;
7              se u ⩽ 0 então
8                  u = |V| − 1;
9              fim
10             V'_jt = V'_jt + g_{n+1} V_{u+1};
11         fim
12         V'_{j(t+1)} = V'_jt;
13     fim
14 fim
```

Primeiramente, a técnica para análise de componentes escolhida é utilizada para calcular as componentes C do conjunto original de dados sensoriados V. Após o cálculo das componentes, a primeira componente C_1 é selecionada e seus respectivos escores são ordenados. Com isso, em função do tipo de escores a ser utilizado, ou seja, os escores maiores, menores ou intermediários, as posições desses escores da componente C_1 são usadas para referenciar as posições das linhas em V que irão compor o conjunto de dados reduzido V', de acordo com o nível de redução n' empregado. Por fim, o conjunto de dados reduzido V', contendo as linhas de V mais representativas para a aplicação, é obtido e posteriormente enviado ao sorvedouro. O pseudo-código é mostrado no algoritmo 4.

- Na linha 2, tem-se o cálculo das componentes, através da técnica escolhida. A ordem de complexidade do cálculo de PCA pode ser estimada em $O(s^2s' + s^2n)$, onde s corresponde ao número de sensores do conjunto de dados original, s' é o número de sensores do conjunto de dados reduzido e n o tamanho da amostra

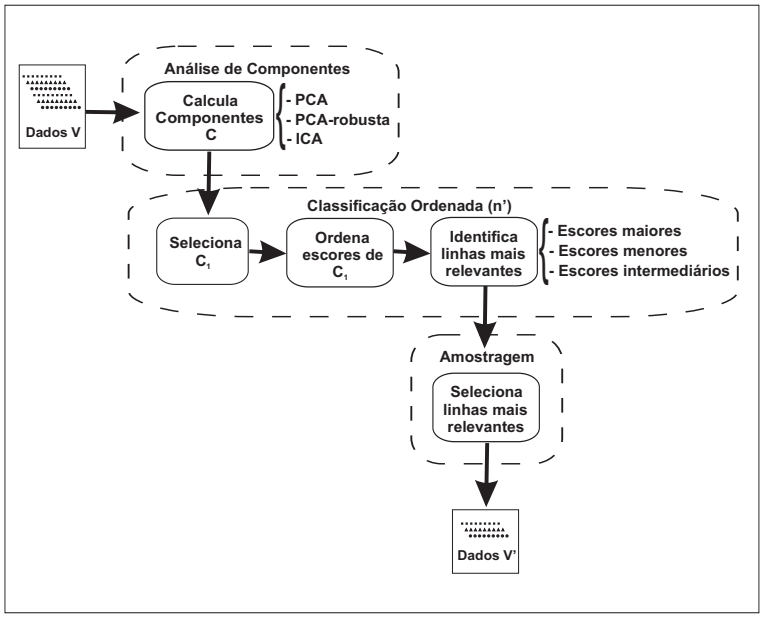


Figura 4.3: Passos do algoritmo baseado em componentes principais

dos dados. Como nesse caso $s = s'$ e $s < n$, tem-se a ordem de complexidade $O(s^2 n)$. Para o cálculo de ICA, considerando o algoritmo FastICA, essa ordem pode ser estimada em $O(sn)$ [85]. A ordem de complexidade de PCA-robusta pode ser estimada em $O(sk^2 n)$ [24], onde k representa o número de componentes principais desejado. Como neste trabalho, apenas a primeira componente principal é necessária, a ordem de complexidade é $O(sn)$.

- Na linha 3, a primeira componente C_1 é selecionada.
- Na linha 4, tem-se a ordenação do vetor com a primeira componente C_1, no qual os índices I dos escores de C_1 são obtidos. A ordem de complexidade da ordenação é $O(n \log_2 n)$, uma vez que $|C_1| = n$.
- Na linha 5, tem-se a ordenação do vetor I, em função do tipo de escore escolhido, considerando apenas os n' primeiros índices.

Algorithm 4: Redução multivariada

```
   Entrada: V – dados de entrada
   Entrada: n' – tamanho da redução
   Entrada: esc – escore a ser utilizado
   Saída: V' – dados reduzidos
1  início
2  |  C ← calculaComponentes(V);
3  |  C_1 ← a primeira componente de C;
4  |  I ← ordena(C_1)  /* Ordena escores de C_1 */ ;
5  |  I ← ordena(I, n', esc)  /* I contm os escores de C_1 */ ;
6  |  para i ← 1 até n' faça
7  |  |  V'_i ← V'_{I_i};
8  |  fim
9  fim
```

Nesse caso, n' representa a quantidade de dados que irá compor o conjunto de dados reduzido V' e I contém os índices dos valores mais relevantes, ou seja, das linhas que contêm os dados mais significativos de V. Isso é necessário para se manter a ordem de chegada dos elementos escolhidos para V'. A ordem de complexidade da ordenação é $O(n' \log_2 n')$.

- Nas linhas 6 – 8, tem-se a montagem dos dados de saída reduzidos V', cuja ordem de complexidade é $O(n')$.

Sendo assim, no caso da utilização de PCA, a complexidade de tempo total é $O(s^2 n) + O(n \log_2 n) + O(n' \log_2 n') + O(n') = O(s^2 n) + O(n \log_2 n)$. Utilizando ICA ou PCA-robusta, a complexidade de tempo total é $O(sn) + O(n \log_2 n) + O(n' \log_2 n') + O(n') = O(sn) + O(n \log_2 n)$.

Para a complexidade de espaço, considere as matrizes V, V', C, que correspondem respectivamente aos dados de entrada, dados de saída e componentes principais ou independentes. A complexidade de espaço é dada por $2\,O(s\,n) + O(s\,n') = O(s\,n)$. Uma vez que cada nó fonte envia V' até o sorvedouro, a complexidade de comunicação é $O(s\,n'\,\rho)$, onde ρ é a maior rota na rede.

4.5 Exercícios de verificação de aprendizagem

1. Quais os principais requisitos da aplicação que nos leva a considerar uma solução baseada em rascunho de dados ou amostragem?
2. Quais as vantagens e desvantagens dos algoritmos baseados em rascunho e em amostragem?
3. Qual a diferença, em relação à qualidade dos dados após a redução, entre os algoritmos de amostragem aleatória e central?
4. Qual propriedade da distribuição dos dados o algoritmo de amostragem central utiliza para melhorar a qualidade dos dados reduzidos? Explique o processo de redução enfatizando tal propriedade.
5. Considerando a complexidade dos algoritmos de amostragem univariada e multivariada, qual seria a carga de processamento, armazenamento e comunicação por hora assumindo que temos cerca 10^6 dados sendo gerados por minuto?

Capítulo 5

Estudo de caso

Este capítulo apresenta três estudos de caso para a redução: no momento do sensoriamento, em redes hierárquicas e durante o roteamento para atender a prazos de aplicações de tempo real. Para todos os casos apresentaremos como o arcabouço, ilustrado anteriormente, pode ser empregado no desenvolvimento das aplicações.

- Para o *sensoriamento*, serão apresentados diferentes casos onde os dados que representam os fenômenos monitorados pelos nós sensores podem ser reduzidos de forma a balancear os gastos sob o ponto de vista da infraestrutura de rede.
- Para a *redução em redes hierárquicas*, será apresentada uma formulação matemática mostrando que as aplicações gerais têm um melhor desempenho quando modeladas para uma rede hierárquica ao invés de plana.
- Para a *redução em aplicações de tempo real*, será apresentada uma formulação matemática para estimar o tamanho de redução ideal no momento do roteamento de tal forma que as aplicações de tempo real possam atender aos prazos sem degradar a qualidade dos dados.

Será mostrado, através de algumas simulações, que por intermédio da redução de dados em redes planas, hierárquicas e no momento do

roteamento, é possível se obter uma considerável economia de recursos da rede sem afetar a qualidade nos dados.

5.1 Redução no momento do sensoriamento

Nesse estudo de caso consideramos a redução de dados univariados em aplicações gerais que monitoram dados básicos, como temperatura, umidade ou luminosidade. Nessas aplicações quanto mais leituras do ambiente tivermos mais precisa e eficiente pode ser a decisão tomada pela aplicação. Alguns exemplos de aplicações gerais que utilizam dados univariados são: monitoramento de incêndios, detecção de enchentes, detector de poluição e agricultura de precisão. De uma forma geral, nesses tipos de aplicações os nós são distribuídos de forma aleatória e os eventos monitorados podem ser enviados para o sorvedouro periodicamente.

As aplicações gerais onde os dados de entrada são dados univariados podem ser modeladas com um arcabouço de redução seguindo as seguintes operações:

- *Caracterização*: Os dados serão modelados como *Univariados*.
- *Suporte a redução*: Os algoritmos utilizados serão os de *amostragem* e *rascunho* definidos no capítulo 4.
- *Robustez*: Utilizamos dois testes estatísticos: um para avaliar se a distribuição dos dados reduzidos é a mesma dos dados originais; e outro para avaliar qual erro agregado ao valor médio dos dados reduzidos quando comparado ao valor médio dos dados originais. Ambos os testes foram definidos na seção 2.3.
- *Concepção*: A forma de apresentar o comportamento da redução dos dados foi via simulação que será apresentada a seguir.

As simulações realizadas são baseadas nas seguintes considerações:

- As simulações foram feitas na ferramenta de simulação NS-2 (*Network Simulator 2*) versão 2.33. Cada cenário simulado foi executado em 33 diferentes topologias aleatórias. Ao fim, para cada

cenário, apresentamos os resultados utilizando a média das 33 execuções com intervalo de confiança de 95%.

- Utilizamos um algoritmo de roteamento baseado em árvore chamado EF-Tree [60]. A densidade da rede é mantida constante e todos os nós têm a mesma configuração de *hardware*. A árvore é construída apenas uma vez na fase de estabelecimento da rede, pois o consumo para montagem da árvore pode interferir nós resultados.

- O parâmetro de configuração variado foi o número de nós monitorando o ambiente.

- Para o algoritmo de *amostragem aleatória* analisamos o comportamento da rede e a qualidade dos dados para cada parâmetro de configuração utilizando um conjunto de amostras de tamanhos $|V|/2$ e $\log_2 |V|$.

Alguns parâmetros importantes utilizados nas simulações são apresentados na tabela 5.1[1].

Tabela 5.1: Parâmetros de simulação para redução de dados univariados.

Parâmetro	**Valor**
Tamanho da rede	varia com a densidade
Tamanho da fila	varia com $\|V\|$
Tempo de simulação (s)	5000
Tráfego (s)	[1000, 4000]
Período do envio (s)	60
Alcance do rádio (m)	50
Largura de banda (kbps)	250
Energia inicial (Joules)	1000
Localização do sorvedouro	coordenadas (0, 0)

Inicialmente considere o comportamento da rede, onde são identificados o consumo total de energia na rede e a média do atraso, para entregar um pacote partindo dos nós que estão monitorando o ambiente até o sorvedouro. Além disso, para comparação dos cenários

[1]Os parâmetros utilizados levam em conta a arquitetura do MicaZ.

utilizamos o comportamento da rede sem utilizar nenhuma redução. As curvas nas figuras representam os resultados para a utilização dos algoritmo *rascunho* e *amostragem aleatória* e o comportamento da rede sem utilizar redução (N).

A figura 5.1 mostra no eixo-y o consumo médio de energia em Joules para a avaliação do comportamento da rede e no eixo-x o número de nós monitorando o ambiente em $1, 5, 10$ e 20. Fixamos o número de nós na rede em 128 e o tamanho dos dados de entrada em 256.

Analisando a figura 5.1 quando a redução diminui o consumo de energia também diminui, onde a amostra-$(\log_2 N)$ e a solução de rascunho apresentam o melhor desempenho. Isso ocorre porque para ambos os casos enviamos para o sorvedouro apenas um pacote com no máximo 20 leituras do ambiente, o que corresponde ao tamanho máximo do pacote considerado.

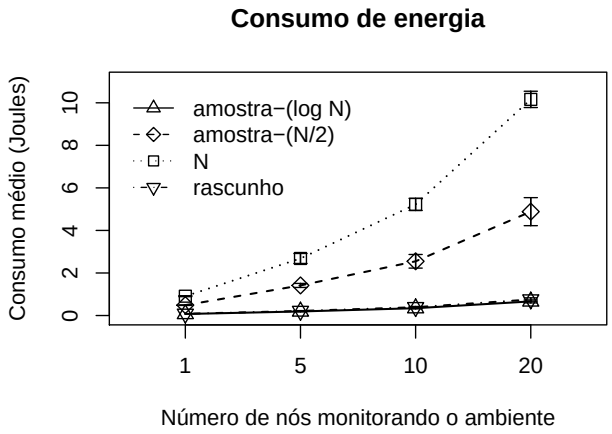


Figura 5.1: Avaliação do comportamento da rede, considerando a média de energia consumida na rede ao reduzir dados univariados.

Em relação ao atraso nos pacotes, temos a figura 5.2 que mostra no eixo-y o atraso médio em segundos e no eixo-x o número de nós monitorando o ambiente em $1, 5, 10$ e 20. Fixamos o número de nós na rede em 128 e o tamanho dos dados de entrada em 256.

Como para o consumo de energia, podemos observar que quando o tamanho dos dados reduzidos diminui temos um menor atraso, pela mesma razão do consumo de energia. Os mesmos efeitos do consumo de energia para número de nós monitorando o ambiente são observados no atraso dos pacotes. Mais uma vez, em todos os casos a amostra-$(\log_2 N)$ e o rascunho tiveram o melhor desempenho.

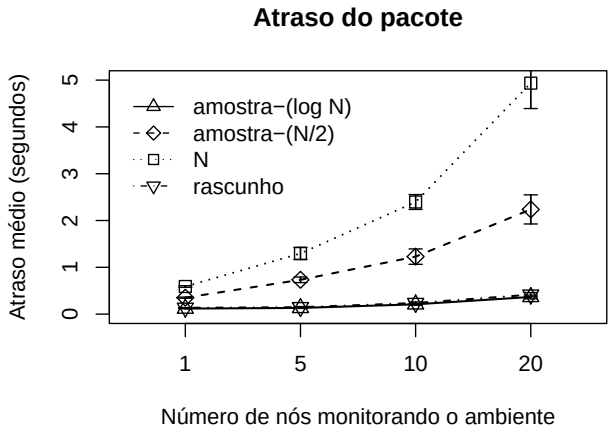


Figura 5.2: Avaliação do comportamento da rede, considerando a média do atraso do pacote ao reduzir dados univariados.

Para analisar o comportamento dos dados reduzidos consideramos a verificação se os dados originais e reduzidos seguem a mesma distribuição através do teste-KS; e o valor absoluto do erro relativo. As curvas nas figuras representam os resultados para a utilização do algoritmo *amostragem aleatória* com tamanhos reduzidos em $\{\log_2 N, N/2\}$ e o comportamento dos dados sem utilizar redução (N).

A figura 5.3 mostra no eixo-y o erro médio em porcentagem[2] ao utilizarmos o teste-KS para a avaliação do comportamento dos dados e no eixo-x o número de nós monitorando o ambiente em $1, 5, 10$ e 20. Fixamos o número de nós na rede em 128 e o tamanho dos dados de entrada em 256.

[2]Como o dado reduzido possui valores entre 0 e 1, podemos considerar o erro em porcentagem.

De forma geral, os valores encontrados para o erro referente à distância vertical do teste KS foram de 20% para a amostra-$(\log_2 N)$ e 10% para a amostra-$(N/2)$. Em todos os casos, o erro é constante porque a perda de pacotes na rede é muito pequena, ou seja, tudo que é enviado chega ao sorvedouro. O maior erro ocorre quando utilizamos a amostra-$(\log N)$, porém a similaridade entre os dados ainda é preservada.

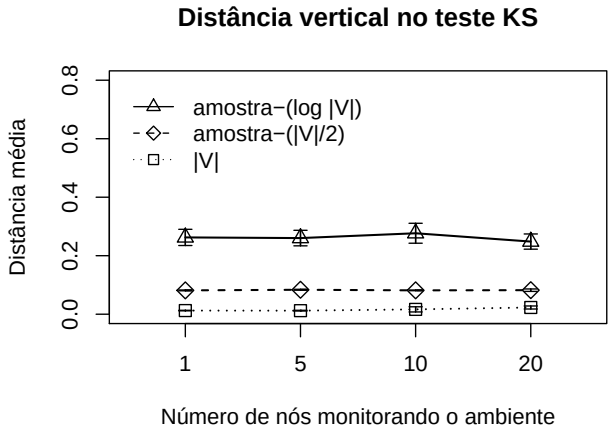


Figura 5.3: Avaliação do comportamento dos dados reduzidos, considerando o erro médio ao aplicar o teste-KS sobre os dados univariados.

A figura 5.4 mostra no eixo-y o erro médio em porcentagem ao utilizarmos a diferença do valor da média para a avaliação do comportamento dos dados e no eixo-x o número de nós monitorando o ambiente em $1, 5, 10$ e 20. Fixamos o número de nós na rede em 128 e o tamanho dos dados de entrada em 256

De forma geral, os valores encontrados para o erro referente aos valores das amostras foram de 10% para a amostra-$(\log_2 N)$ e praticamente 0% para a amostra-$(N/2)$. Como na avaliação do teste-KS, o erro é constante porque a perda de pacotes na rede é muito pequena, ou seja, tudo que é enviado chega ao sorvedouro. Com isso, se quisermos manter o máximo da qualidade do dado considerando apenas o

erro médio podemos utilizar a amostra-$N/2$.

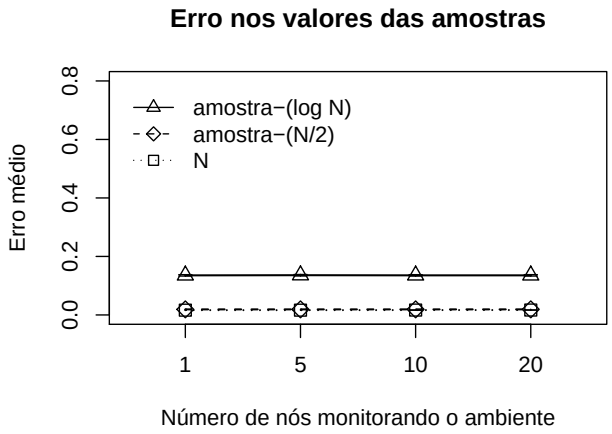


Figura 5.4: Avaliação do comportamento dos dados reduzidos, considerando o erro médio sobre os dados univariados.

5.2 Redução em redes hierárquicas

Nesse estudo de caso consideramos a redução de dados univariados em aplicações gerais que monitoram dados básicos, como temperatura, umidade ou luminosidade. Porém, a topologia da rede é hierárquica, ou seja, nós com baixo poder de processamento monitoram o ambiente e nós mais poderosos recebem, reduzem e propagam os dados para o sorvedouro.

As aplicações gerais em redes hierárquicas onde os dados de entrada são dados univariados podem ser modeladas com um arcabouço de redução seguindo as seguintes operações:

- *Caracterização*: Os dados serão modelados como *Univariados*. E a viabilidade de se utilizar redução de dados nessa rede é caracterizada.

- *Suporte a redução*: Os algoritmos utilizados serão os de *amostragem* e *rascunho* definidos no capítulo 4.

- *Robustez*: Utilizamos dois testes estatísticos: um para avaliar se a distribuição dos dados reduzidos é a mesma dos dados originais; e outro para avaliar qual erro agregado ao valor médio dos dados reduzidos quando comparado ao valor médio dos dados originais. Ambos os testes foram definidos na seção 2.3.

- *Concepção*: A forma de apresentar o comportamento da redução dos dados foi via simulação que será apresentada a seguir.

5.2.1 Redes planas vs. redes hierárquicas

Para mostrar que o desempenho em número de bits transmitidos das redes hierárquicas com redução é melhor que o desempenho das redes planas sem redução, pode ser utilizado o modelo analítico proposto por Vlajic et al. [78]. Isso foi feito comparando o custo operacional em bits transmitidos de cada uma das redes. Para obter o custo operacional das redes planas sem efetuar redução, deve-se fazer as seguintes considerações:

- Existem $s = |S|$ nós sensores na rede.

- Os nós são distribuídos uniformemente sobre uma área $Area = L \times L$ dividida em grade, onde cada célula da grade com tamanho $a \times a$ contém apenas um nó. Além disso, o sorvedouro está localizado no centro da área.

- Cada nó comunica-se apenas com os oito nós localizados nas células vizinhas.

- $\overline{d}$ representa a média da distância entre os nós e o sorvedouro. Como estamos considerando os nós dispostos numa área quadrada dividida em grade, Vlajic et al. [78] mostram que temos, em número de saltos, $\overline{d} = \sqrt{s}/3$.

- Cada nó envia um conjunto de tamanho $|V|$ para o sorvedouro.

Baseado nessas considerações o número de bits (T_p) transmitidos numa rede plana sem redução é definido como

$$T_p = \overline{d}\,|V|\,s. \tag{5.2.1}$$

Para derivar o custo operacional de uma rede hierárquica temos as seguintes considerações:

- Existem $a = |A|$ agrupamentos na rede.
- A área da rede é dividida em zonas quadradas fixas, iguais e não sobrepostas de tamanho $L/a \times L/a$, onde cada zona quadrada contém um líder no centro. O sorvedouro está localizado no centro da área.
- Como a representa o número de agrupamentos, s/a é o número de nós por agrupamento.
- Segundo Vlajic et al. [78], a média da distância em saltos dentro do agrupamento é $\overline{d_a} = \overline{d}/\sqrt{a}$.
- O fator médio de redução de dados dentro do agrupamento por cada leitura de sensoriamento reportada é dada por α e a quantidade média de dados enviados para o sorvedouro é $\alpha\,|V|\,s/a$.

Baseado nessas considerações o número de bits (T_h) transmitidos numa rede hierár-quica é definido como

$$T_h = a\Big(\frac{s}{a} - 1\Big)\overline{d_a}\,|V| + a\,\overline{d}\Big(\alpha\,|V|\,\frac{s}{a}\Big), \tag{5.2.2}$$

onde o primeiro termo da soma representa a quantidade de bits que trafegam dentro dos agrupamentos e o segundo a quantidade de bits que trafegam na rede até o sorvedouro. Simplificando e substituindo $\overline{d_a}$ por $\overline{d}/\sqrt{a}$ na equação (5.2.2) temos

$$T_h = (s-a)\frac{\overline{d}}{\sqrt{a}}\,|V| + a\,\overline{d}\Big(\alpha\,|V|\,\frac{s}{a}\Big). \tag{5.2.3}$$

Baseado nas equações (5.2.1) e (5.2.3), Vlajic & Xia [78] mostram que

$$T_h < T_p \iff \frac{s-a}{\sqrt{a}} + \alpha\, s < s, \tag{5.2.4}$$

ou seja, a quantidade de bits transmitidos pela rede hierárquica com redução só será inferior a quantidade de bits transmitidos pela rede plana sem redução se a inequação acima for satisfeita.

Para efetuarmos a redução de dados nos nossos cenários de aplicações gerais de redução, devemos considerar $|V'| = \alpha\,|V|\,s/a$. Com isso,

$$T_h = (s-a)\frac{\overline{d}}{\sqrt{a}}\,|V| + a\,|V'|\overline{d}, \tag{5.2.5}$$

onde $|V'|$ é a quantidade de dados reduzidos no agrupamento e enviados até o sorvedouro. Com isso, para os nossos cenários de redução, a inequação (5.2.4) pode ser reescrita da seguinte forma

$$T_h < T_p \iff \frac{s-a}{\sqrt{a}} + \frac{|V'|a}{|V|} < s. \tag{5.2.6}$$

Como $a < s$, a inequação acima sempre será satisfeita, ou seja, o consumo em bits transmitidos numa rede hierárquica com redução sempre será inferior à quantidade de bits transmitidos na rede plana sem redução. No entanto essa formulação pode ser utilizada para encontrarmos o número ideal de agrupamentos numa aplicação de redução de dados.

Considere outro cenário, um pouco mais real, onde a RSSF possui $s = 160$ nós, $a = (4, 9, 16, 25)$ agrupamentos, cada nó do agrupamento gerando um item com o tamanho $|V| = 256$ e o líder fazendo reduções $|V'| \in \{\log|V|, |V|/2, |V|\}$. Mais uma vez em todos os casos, através das equações (5.2.1) e (5.2.5), calculamos a razão T_h/T_p, que nos levam aos resultados apresentados na tabela 5.2. Como nos resultados anteriores, em todos os casos a redução de dados na rede hierárquica possui um melhor desempenho quando comparada a uma rede plana sem redução.

Como pode ser visto, as redes hierárquicas com redução possuem um melhor desempenho quando comparadas às redes planas.

Tabela 5.2: Razão de bits transmitidos numa rede com 160 nós.

Agrupamentos	$\log \lvert V \rvert$	$\lvert V \rvert / 2$	$\lvert V \rvert$
4	0.48	0.50	0.51
9	0.33	0.36	0.38
16	0.22	0.27	0.32
25	0.14	0.24	0.34

5.2.2 Simulações

As simulações realizadas para a avaliação do comportamento das redes hierárquicas são baseadas nas mesmas apresentadas na seção 5.1.

Inicialmente considere a avaliação do comportamento da rede hierárquica nas aplicações gerais. Para essa avaliação identificamos o consumo total de energia na rede e a média do atraso para entregar um pacote dos líderes até o sorvedouro. Além disso, utilizamos para comparação os resultados referentes à eficiência, sem utilizar nenhuma solução de redução ($|V|$ amostras). As curvas nas figuras 5.5 e 5.6 representam os resultados para a utilização dos algoritmo *rascunho* e *amostragem aleatória* com $|V'| \in \{\log |V|, |V|/2\}$ e o comportamento da rede sem utilizar redução ($|V|$).

A figura 5.5 mostra no eixo-y o consumo médio de energia em Joules para a avaliação do comportamento da rede e no eixo-x o número agrupamentos em $4, 8, 12$ e 16. Para todos os cenários, fixamos o número de nós no agrupamento em 100, o tamanho do conjunto de dados identificado no líder em 256.

Analisando de forma geral, como esperado, em todos os casos para o *amostragem aleatória* observamos que quando $|V'|$ diminui, o consumo de energia também diminui. Para o *rascunho* observamos um comportamento similar ao resultado para a amostra-($\log |V|$). Além disso, esses resultados estão de acordo com a análise feita na seção 5.2.1, pois quanto maior $|V'|$ melhor é o desempenho da rede hierárquica. Vale destacar que o consumo de energia, é maior, quando temos poucos agrupamentos. Isso ocorre porque o tráfego dentro do agrupamento é grande, ou seja, o número de agrupamentos ainda não é o ideal, como sugerido na formulação da seção 5.2.1.

A figura 5.6 mostra no eixo-y o atraso médio em segundos dos

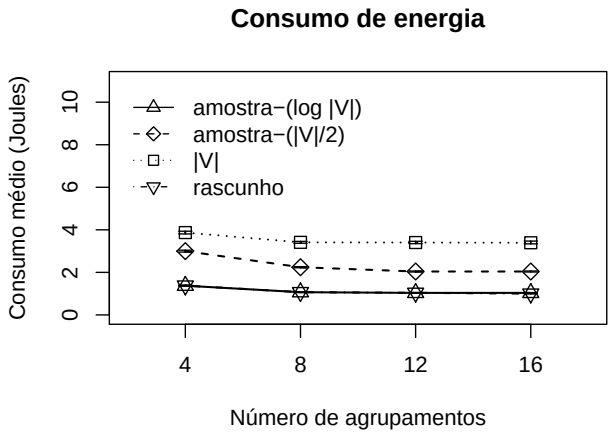


Figura 5.5: Avaliação do comportamento da rede, considerando a média da energia consumida na rede ao reduzir dados nos nós líderes.

pacotes e no eixo-x o número agrupamentos em $4, 8, 12$ e 16. Para todos os cenários, fixamos o número de nós no agrupamento em 100, o tamanho do conjunto de dados identificado no líder em 256.

Como nos resultados para o consumo de energia, podemos observar que quando o $|V'|$ diminui temos um menor atraso.

Ao considerar a avaliação do comportamento dos dados reduzidos nas redes hierárquicas, é observado o mesmo comportamento da seção 5.1 referente a redução de dados univariados no momento do sensoriamento. Por essa razão tal avaliação será omitida aqui.

5.3 Redução em aplicações de tempo real

Nesse estudo de caso consideramos a redução de dados univariados em aplicações de tempo real que monitoram dados que exigem um prazo para serem entregues, como alarme de incêndio, vazamento em uma usina ou risco de explosão. As aplicações de tempo real cujos os dados de entrada são dados univariados podem ser modeladas com um arcabouço de redução seguindo as seguintes operações:

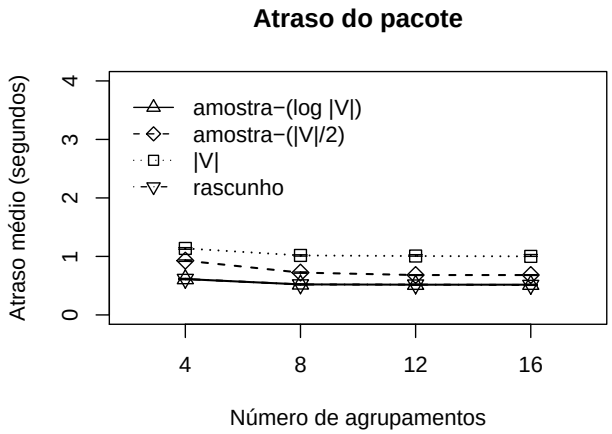


Figura 5.6: Avaliação do comportamento da rede, considerando a média do atraso do pacote ao reduzir dados nos nós líderes.

- *Caracterização*: Os dados serão modelados como *Univariados*. E os requisitos da rede para a obtenção dos prazos devem ser caracterizados.
- *Suporte a redução*: O algoritmos utilizado é o de *amostragem*.
- *Robustez*: Utilizamos dois testes estatísticos: um para avaliar se a distribuição dos dados reduzidos é a mesma dos dados originais; e outro para avaliar qual erro agregado ao valor médio dos dados reduzidos quando comparado ao valor médio dos dados originais. Ambos os testes foram definidos na seção 2.3.
- *Concepção*: A forma de apresentar o comportamento da redução dos dados foi via simulação que será apresentada a seguir.

5.3.1 Caracterização para a obtenção dos prazos

O principal requisito a ser caracterizado é o de como determinar, nos nós roteadores na fase de propagação, o tamanho do conjunto V' de tal

forma que os prazos especificados pela aplicação sejam atendidos. Com isso, para estimar o tamanho de V' faremos as seguintes considerações:

- Para identificar o atraso dos dados consideramos que todos os nós estão com os relógios sincronizados.
- Cada sensor assume um tamanho máximo para o pacote (pct_t), no nosso caso, $pct_t = 20$.
- Em cada nó roteador, um novo prazo local (p_l), entre o nó roteador e o sorvedouro, é calculado da seguinte forma:

  $$p_l = p_a - t_{atual},$$

  onde t_{atual} é o tempo atual do sistema.
- O tempo estimado (t_{org}) que o fragmento V^j gasta para viajar entre o nó origem e o nó roteador é

  $$t_{org} = t_{atual} - t_{gen}.$$

- Considere t_{dst} o tempo que o fragmento V^1 gasta para viajar do nó roteador até o sorvedouro. O fragmento V^2 gastará o tempo t_{dst}/s_{dst} para viajar entre o nó roteador e o sorvedouro e o tempo t_{org}/s_{org} para viajar entre o nó origem e o nó roteador. Dessa forma, o tempo estimado para entrega e recebimento de V é respectivamente:

  $$t_{ent} = t_{dst} + (n_f - 1)\, t_{dst}/s_{dst} \tag{5.3.7}$$

  e

  $$t_{rec} = (n_f - 1)\, t_{org}/s_{org}. \tag{5.3.8}$$

  O primeiro termo da soma não é considerado para t_{rec} porque V^1 já chegou.

Baseado nessas considerações, $|V'|$ é determinado e usado somente se $gap > 0$, onde o gap é dado por:

$$gap = p_l - atraso, \tag{5.3.9}$$

com o *atraso* estimado para entregar V, após receber o primeiro fragmento V^1, até o sorvedouro sendo

$$atraso = t_{rec} + t_{ent}. \tag{5.3.10}$$

Sendo o $gap > 0$, de (5.3.9) e (5.3.10) temos

$$p_l - atraso > 0$$

$$p_l - (t_{rec} + t_{ent}) > 0$$

usando (5.3.8) e (5.3.7) nós temos

$$p_l - ((n_f - 1)\, t_{org}/s_{org} + t_{dst} + (n_f - 1)\, t_{dst}/s_{dst}) > 0$$

simplificando

$$n_f < 1 + \frac{s_{org}\, s_{dst}\, (p_l - t_{dst})}{s_{dst}\, t_{org} + s_{org}\, t_{dst}}$$

considerando que $n_f = \lceil |V'|/pct_t \rceil$, temos

$$|V'| < pct_t \left(1 + \frac{s_{org}\, s_{dst}\, (p_l - t_{dst})}{s_{dst}\, t_{org} + s_{org}\, t_{dst}}\right).$$

Finalmente para atender a desigualdade temos

$$|V'| = pct_t \left(1 + \frac{s_{org}\, s_{dst}\, (p_l - t_{dst})}{s_{dst}\, t_{org} + s_{org}\, t_{dst}}\right) - 1. \tag{5.3.11}$$

5.3.2 Simulações

As simulações realizadas para a avaliação do comportamento das aplicações de tempo real, ao utilizar a redução na camada de roteamento, são baseadas nas seguintes considerações:

- A avaliação foi realizada na ferramenta de simulação NS-2 (*Network Simulator 2*) versão 2.33. Cada cenário simulado foi executado em 33 diferentes topologias aleatórias. Ao fim, para cada cenário apresentamos os resultados utilizando a média das 33 execuções com intervalo de confiança de 95%.

- Foi utilizado um algoritmo de roteamento baseado em árvore EF-Tree. A densidade da rede é mantida constante e todos os nós têm a mesma configuração de *hardware*. A árvore é construída apenas uma vez na fase de estabelecimento da rede, pois o consumo para montagem da árvore pode interferir nos resultados. Além disso, os nós são distribuídos numa $Area = L \times L$ com o sorvedouro localizado na posição $(0, 0)$ e um único nó monitorando o ambiente localizado na posição (L, L).

- Utilizamos uma aplicação geral com exigências de prazos para a entrega do item V. Para utilizar prazos mais realistas, determinamos de forma empírica o prazo mínimo (p_{min}) exigido em cada cenário simulado. Com isso, utilizamos nas nossas avaliações os seguintes prazos da aplicação: $p_a = \frac{p_{min}}{2}$ com a rede funcionando sem atraso; e $p_a = p_{min}$ com cada nó roteador na rede gerando um $atraso = \frac{p_{min}}{10000}$ em todos os pacotes recebidos, ou seja, cada nó só repassa cada pacote recebido por $atraso$ segundos.

- O parâmetro de configuração variado foi o tamanho do conjunto de dados em bytes.

- Para o algoritmo de *amostragem aleatória* analisamos o comportamento da rede e a qualidade dos dados para cada parâmetro de configuração utilizando um conjunto de amostras de tamanhos $|V|/2$ e $\log_2 |V|$.

Os demais parâmetros são os mesmos apresentados na na seção 5.1.

Para avaliação foi considerado um cenários específicos com tráfego. Avaliamos o comportamento da solução ao utilizarmos prazos da aplicação $p_a = p_{min}$ com os nós atrasando os pacotes. Além disso, uma questão importante a ser considerada ao determinar o prazo da aplicação é o prazo mínimo (p_{min}) permitido pela rede para entregar o item V ao sorvedouro. O valor p_{min} pode ser difícil de se determinar, pelas condições dinâmicas durante a operação da rede, tais como número de nós fontes, quantidade de dados e topologias. Por essas razões, os nossos cenários são avaliados utilizando como base os valores para p_{min} obtidos de forma empírica.

Além dos nós da aplicação, existem outros nós monitorando alguma variável do ambiente e enviando suas leituras ao sorvedouro, gerando assim tráfego concorrente. Nas avaliações, consideramos uma rede com 128 nós, apenas um nó monitorando o ambiente, variamos o tamanho do dos dados $|V|$ em 256, 512, 1024 e 2048 e para identificarmos o impacto da nossa solução numa rede com tráfego utilizamos 16%, 20%, 25% e 33% dos nós gerando tráfego até o sorvedouro. Seguindo as mesmas estratégias do cenário I, os valores de p_{min} são as médias dos tempos gastos para V ser inteiramente recebido pelo sorvedouro. A tabela 5.3 resume os valores de p_{min} para os diferentes tamanhos $|V|$ em cada porcentagem de nós monitorando o ambiente. Os valores para p_{min} com os intervalos de confiança são ilustrados na figura 5.7, onde no eixo-x temos a variação do tamanho do conjunto de dados, no eixo-y temos a média do atraso em segundos e as colunas representam a porcentagem dos nós na rede gerando tráfego.

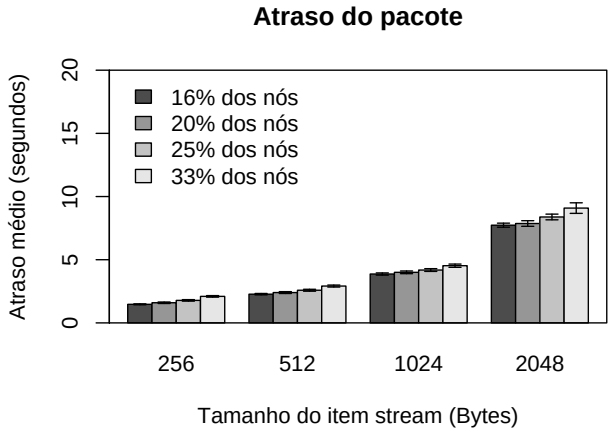


Figura 5.7: Valores mínimos para os prazos exigidos pelas aplicações.

É importante destacar que, se a aplicação tem um prazo menor que os mostrado na tabela 5.3, os dados não poderão ser entregues no prazo e alguma redução é necessária.

Tabela 5.3: Valores mínimos para os prazos exigidos pelas aplicações.

Porcentagem dos nós	**Tamanho do conjunto de dados**			
	256	512	1024	2048
16%	1.47s	2.27s	3.86s	7.73s
20%	1.59s	2.40s	4.00s	7.86s
25%	1.78s	2.58s	4.18s	8.38s
33%	2.09s	2.91s	4.52s	9.08s

Considere o prazo da aplicação $p_a = p_{min}$ com os nós roteadores atrasando os fragmentos de dados por $\frac{p_{min}}{10000}$ ou 0.01% do valor de p_{min}. A tabela 5.4 mostra os resultados do *atraso* para essa avaliação. Os valores para o *atraso* com os intervalos de confiança são ilustrados na figura 5.8, onde no eixo-x temos a variação do tamanho do conjunto de dados, no eixo-y temos a média do atraso, em segundos, e as colunas representam a porcentagem dos nós na rede gerando tráfego.

Tabela 5.4: Atrasos identificados ao utilizar atrasos gerados pelos nós roteadores.

Porcentagem dos nós	**Tamanho do conjunto de dados**			
	256	512	1024	2048
16%	1.28s	1.85s	3.01s	5.14s
20%	1.37s	1.89s	3.01s	5.32s
25%	1.53s	2.01s	2.94s	5.54s
33%	1.81s	2.21s	3.05s	5.78s

Como esperado, a tabela 5.4 mostra que p_a é alcançado em todos os casos, quando comparados com os prazos apresentados na tabela 5.3. Isso acontece, pois o tamanho estimado para a redução (equação (5.3.11)) está diretamente relacionado ao prazo da aplicação p_a.

Ao avaliar a qualidade dos dados, observamos que a redução efetuada mantem uma boa representatividade dos dados sendo assim permissível para um conjunto de aplicações. Os resultados para os erros, com os intervalos de confiança, podem ser vistos na figura 5.9, onde no eixo-x temos a variação do tamanho do conjunto de dados, no eixo-y temos o erro médio e as colunas representam a porcentagem dos nós na rede gerando tráfego.

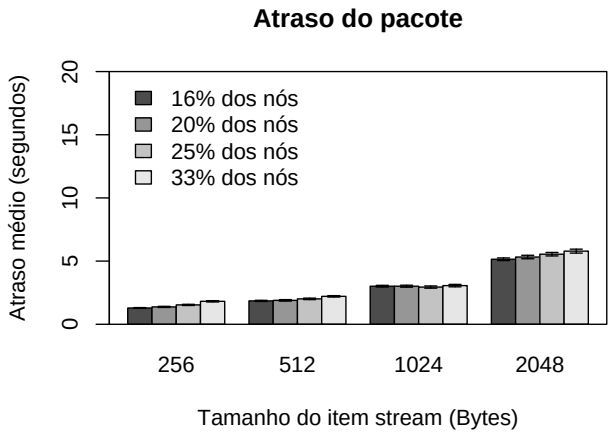


Figura 5.8: Atrasos identificados ao utilizar atrasos gerados pelos nós roteadores.

5.4 Exercícios de verificação de aprendizagem

1. Como seria a utilização de redução de dados multivariada no cenário de redução no momento do sensoriamento? Apresente uma caracterização detalhada.

2. Apesar do algoritmo de rascunho apresentar um consumo baixo de energia, qual a sua desvantagem em relação ao algoritmo de amostragem quando consideramos a qualidade dos dados?

3. Apresente uma lista de aplicações onde o algoritmo de rascunho possa ser utilizado sem causar prejuízo na qualidade dos dados.

4. Considerando a utilização de redução de dados em redes hierárquicas, apresente um cenário onde a divisão da rede em agrupamentos efetuando a redução não é viável. Utilize como ponto de partida a tabela 5.2.

5. Como seria a utilização de redução de dados multivariada no cenário de redução em redes hierárquicas? Apresente uma caracterização detalhada.

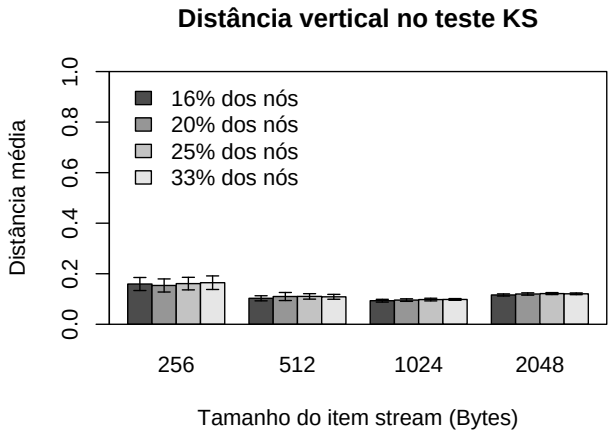


Figura 5.9: Erros identificados ao utilizar atrasos gerados pelos nós roteadores.

6. Se utilizarmos o algoritmo de amostragem baseado em *Wavelets* no cenário de tempo real, é possível considerarmos a qualidade dos dados combinada com o requisito de tempo? Caso seja possível, apresente uma discussão sucinta de como isso pode ser considerado. Caso contrário, justifique.

Capítulo 6

Considerações finais

As RSSFs possuem diversas limitações, dentre elas a quantidade de energia disponível nas baterias de cada nó sensor. Assim, aumentar seu tempo de vida é uma das tarefas mais importantes no projeto dessas redes. Normalmente, as RSSFs coletam uma grande quantidade de dados do ambiente que precisam ser filtrados, de forma a obter apenas as informações relevantes e/ou descartar dados repetidos ou desnecessários, além de evitar o desperdício de energia no envio desses dados.

Neste livro apresentamos a redução de dados em RSSFs por intermédio de diversas técnicas. Além disso, abordamos um arcabouço para redução de dados que possui uma API para realizar a redução, que pode ser utilizada em diversas aplicações em redes planas, hierárquicas e em aplicações de tempo real. Dentre os principais pontos abordados no livro podemos destacar:

1. A retrospectiva dos principais estudos relacionados a redução de dados em RSSFs.

2. A apresentação de um arcabouço para redução de dados que pode ser utilizado em diversos cenários e aplicações de RSSFs.

3. A apresentação de um conjunto de algoritmos de redução que podem prover suporte ao arcabouço e que permite reduzir os dados nas RSSFs de forma autônoma.

No nosso estudo de caso, quando analisamos a qualidade dos dados e o comportamento da rede, em aplicações gerais utilizando um arcabouço como base, temos as seguintes considerações:

- O *rascunho* quando utilizado apresenta um baixo consumo de energia e atraso pois envia apenas um pacote para o sorvedouro. Uma vez que os dados podem ser gerados "artificialmente" no sorvedouro, a qualidade dos dados não é afetada em relação avaliações estatísticas utilizadas. O problema é a ordem de chegada dos dados que é perdida. Porém, a perda na sequência é aceitável em aplicações onde as restrições da rede são maiores.
- A *amostragem*, quando utilizado com amostra-($\log_2 N$), possui um baixo consumo de energia e atraso pois, para os cenários avaliados, apenas um pacote é enviado para o sorvedouro. Entretanto, a qualidade dos dados é bastante afetada nos testes estatísticos utilizados com erros, respectivamente, de 20% e 10%. Porém, assim como no *rascunho*, essa perda na qualidade pode ser aceitável em aplicações onde as restrições da rede são maiores.
- A *amostragem* com amostra-($N/2$) pode ser utilizado nos casos onde a prioridade da aplicação de sensoriamento é reduzir o erro quando avaliamos o erro médio (no nosso caso perto de zero) ou nos cenários descritos pela figura 5.1, em que o tamanho dos dados de entrada e o número de nós monitorando o ambiente não variam.
- Não é interessante usar redução, ou seja, enviar todos os dados, quando a prioridade da aplicação geral seja reduzir o erro quando aplicamos o teste-KS ou em casos que não existam fortes restrições de rede.

Finalmente, *quando usar amostragem ou rascunho?* Se a ordem dos dados for importante, podemos utilizar amostragem. Nesse caso, devemos sempre analisar os requisitos da aplicação para decidir qual o melhor tamanho do conjunto de amostras. Caso a sequência não seja importante podemos utilizar o rascunho pois ele sempre terá melhor desempenho de rede e ainda manterá a qualidade dos dados quando

aplicarmos os testes estatísticos. Note que essas observações são melhores analisadas ao utilizarmos um arcabouço para guiar essas aplicações gerais.

As aplicações gerais em RSSFs podem utilizar a solução proposta pelo arcabouço, tanto para projetar suas aplicações, como para efetuar a redução em casos onde os nós sensores ou agregadores são os responsáveis por essa tarefa, bem como nos casos que se necessite reduzir os dados durante o roteamento, como nas aplicações de tempo real.

Bibliografia

[1] D. J. Abadi, W. Lindner, S. Madden, e J. Schuler. An integration framework for sensor networks and data stream management systems. In *30th International Conference on Very Large Data Bases (VLDB'04)*, volume 30, pages 1361–1364, Toronto, Canada, September 2004. Morgan Kaufmann.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, e E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40 (8):102–114, August 2002.

[3] C. Alippi, G. Anastasi, C. G. F. Mancini, e M. Roveri. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. In *4th IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS'07)*, pages 1–6, Pisa, Italy, October 2007. IEEE Computer Society.

[4] A. L. L. Aquino. A framework for sensor stream reduction in wireless sensor networks. In *The Fifth International Conference on Sensor Technologies and Applications*, pages 30–35, Nice/Saint Laurent du Var, France, August 2011.

[5] A. L. L. Aquino, R. da S. Cabral, e A. O. Fernandes. Um algoritmo de redução de dados para aplicações de tempo real em redes de sensores sem fio. In *26st Brazilian Symposium on Computer Networks (SBRC'08)*, Rio de Janeiro, Brazil, May 2008. SBC.

[6] A. L. L. Aquino, C. M. S. Figueiredo, E. F. Nakamura, L. S. Buriol, A. A. F. Loureiro, A. O. Fernandes, e C. N. C. Junior. Data stream based algorithms for wireless sensor network applications. In *21st IEEE International Conference on Advanced Information Networking and Applications (AINA'07)*, pages 869–876, Niagara Falls, Canada, May 2007. IEEE Computer Society.

[7] A. L. L. Aquino, C. M. S. Figueiredo, E. F. Nakamura, L. S. Buriol, A. A. F. Loureiro, A. O. Fernandes, e C. N. C. Junior. A sampling data stream algorithm for wireless sensor networks. In *IEEE International Conference on Communications (ICC'07)*, pages 3207–3212, Glasgow, Scotland, June 2007. IEEE Computer Society.

[8] A. L. L. Aquino, C. M. S. Figueiredo, E. F. Nakamura, A. C. Frery, A. A. F. Loureiro, e A. O. Fernandes. Sensor stream reduction for clustered wireless sensor networks. In *23rd ACM Symposium on Applied Computing 2008 (SAC'08)*, pages 2052–2056, Fortaleza, Brazil, March 2008. ACM.

[9] A. L. L. Aquino, C. M. S. Figueiredo, E. F. Nakamura, A. A. F. Loureiro, A. O. Fernandes, e C. N. C. Junior. On the use data reduction algorithms for real-time wireless sensor networks. In *IEEE Symposium On Computers and Communications (ISCC'07)*, pages 583–588, Aveiro, Potugal, July 2007. IEEE Computer Society.

[10] A. L. L. Aquino, A. A. F. Loureiro, A. O. Fernandes, e R. A. F. Mini. An in-network reduction algorithm for real-time wireless sensor networks applications. In *Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP'08)*, Vancouver, British Columbia, Canada, October 2008. ACM.

[11] A. L. L. Aquino e E. F. Nakamura. Data centric sensor stream reduction for real-time applications in wireless sensor networks. *Sensors Basel*, 9:9666–9688, 2009.

[12] A. L. L. Aquino, E. F. Nakamura, A. A. F. Loureiro, e C. J. N. Coelho Jr. Semi-automatic generation of monitoring applications for wireless networks. In *9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'03)*, pages 506–511, Lisbon, Portugal, September 2003. IEEE Computer Society.

[13] A. L. L. Aquino, E. F. Nakamura, A. A. F. Loureiro, e C. J. N. C. Junior. Beanwatcher: a tool to generate multimedia monitoring applications for wireless sensor networks. In A. Marshall e N. Agoulmine, editors, *6th IFIP/IEEE Management of Multimedia Networks and Services (MMNS'03)*, volume 2839 of *Lecture Notes in Computer Science*, pages 128–141, Belfast, Northern Ireland, September 2003. Springer.

[14] A. L. L. Aquino, R. A. R. Oliveira, e E. F. Wanner. A wavelet-based sampling algorithm for wireless sensor networks applications. In *25th ACM Symposium On Applied Computing (SAC 2010)*, pages 1604–1608, Sierra, Switzerland, March 2010. ACM, New York: ACM.

[15] T. Arampatzis, J. Lygeros, e S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *13th IEEE Mediterranean Conference on Control and Automation (MED'05)*, pages 719–724, Hawaii, USA, June 2005. IEEE Computer Society.

[16] S. Brown e C. J. Sreenan. A study on data aggregation and reliability in managing wireless sensor networks. In *4th IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS'07)*, pages 1–6, Pisa, Italy, October 2007. IEEE Computer Society.

[17] T. H. Chan, C. K. Ki, e H. Ngan. Real-time support for wireless sensor networks. technical report. Technical Report RTWSNReport, Hong Kong University of Science and Technology, Hong Kong, JP, September 2005.

[18] V. Chatzigiannakis e S. Papavassiliou. Diagnosing anomalies and identifying faulty nodes in sensor networks. *IEEE Sensors Journal*, 7(5):637–645, May 2007.

[19] B. Chen, K. Jamieson, H. Balakrishnan, e R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, September 2002.

[20] M. Chen, T. Know, e Y. Choi. Energy-efficient differentiated directed diffusion (eddd) in wireless sensor networks. *Computer Communications*, 29(2):231–245, January 2006.

[21] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, April 1994.

[22] N. Cvejic, D. Bull, e N. Canagarajah. Improving fusion of surveillance images in sensor networks using independent component analysis. *IEEE Transactions on Consumer Electronics*, 53 (3):1029–1035, August 2007.

[23] K. Dasgupta, K. Kalpakis, e P. Namjoshi. Improving the lifetime of sensor networks via intelligent selection of data aggregation trees. In *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'03)*, volume 3397 of *Lecture Notes in Computer Science*, pages 508–517, Orlando, Florida, USA, January 2003. Springer.

[24] F. De la Torre e M. J. Black. Robust principal component analysis for computer vision. volume 1, pages 362–369, July 2001.

[25] D. Estrin, L. Girod, G. Pottie, e M. Srivastava. Instrumenting the world with wireless sensor networks. In *26th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, volume 4, pages 2033–2036, Salt Lake City, Utah, USA, June 2001. IEEE Computer Society.

[26] D. Estrin, R. Govindan, J. Heidemann, e S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *5th ACM/IEEE International Conference on Mobile Computing and Networks (MOBICOM'99)*, pages 263–270, Seattle, Washington, USA, August 1999. ACM.

[27] C. M. S. Figueiredo, A. L. L. Aquino, A. A. F. Loureiro, e L. B. Ruiz. Um esquema de gerenciamento para redes de sensores sem fio auto-organizáveis: Atuando sobre regras locais. In *25st Brazilian Symposium on Computer Networks (SBRC'07)*, pages 1–12, Belém, PA, Brazil, May 2007. SBC.

[28] C. M. S. Figueiredo, A. L. dos Santos, A. A. F. Loureiro, e J. M. Nogueira. Policy-based adaptive routing in autonomous wsns. In *16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'05)*, volume 3775 of *Lecture Notes in Computer Science*, pages 206–219, Barcelona, Spain, October 2005. Springer.

[29] A. C. Frery, H. Ramos, J. Alencar-Neto, e E. F. Nakamura. Error estimation in wireless sensor networks. In *23rd ACM Symposium on Applied Computing 2008 (SAC'08)*, pages 1923–1927, Fortaleza, Brazil, March 2008. ACM.

[30] D. Ganesan, S. Ratnasamy, H. Wang, e D. Estrin. Coping with irregular spatio-temporal sampling in sensor networks. *ACM SIGCOMM Computer Communication Review*, 34(1):125–130, January 2004.

[31] B. Gedik, L. Liu, e P. S. Yu. Asap: An adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(12):1766–1783, December 2007.

[32] O. Goussevskaia, M. do V. Machado, R. A. F. Mini, A. A. F. Loureiro, G. R. Mateus, e J. M. Nogueira. Data dissemination

based on the energy map. *IEEE Communications Magazine*, 43 (7):134–143, July 2005.

[33] D. L. Guidoni, A. L. L. Aquino, R. da Silva Cabral, A. A. F. Loureiro, e A. O. Fernandes. Sistemas do tipo eixo-raio aplicados à redes de sensores sem fio modeladas como redes small world. In *39th Brazilian Symposium on Operational Research (SBPO'07)*, pages 1–12, Fortaleza, CE, Brasil, August 2007. SOBRAPO.

[34] A. Guitton, A. Skordylis, e N. Trigoni. Utilizing correlations to compress time-series in traffic monitoring sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'07)*, pages 2479–2483, Las Vegas, USA, April 2007. IEEE Computer Society.

[35] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(1): 417–441, 498–520, 1933.

[36] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, May 1999.

[37] A. Hyvarinen. Survey on independent component analysis. *Neural Computing Surveys*, 2(1):94–128, April 1999.

[38] A. Hyvarinen, J. Karhunen, e E. Oja. *Independent component analysis*. John Wiley & Sons, New York, 1 edition, May 2001. ISBN 047140540X.

[39] A. Hyvarinen e E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, October 1997.

[40] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, e F. Silva. Directed diffusion for wireless sensor networking.

IEEE/ACM Transactions on Networking, 11(1):2–16, February 2003.

[41] J. E. Jackson. *A user's guide to principal components*. Wiley-Interscience, 1 edition, September 2003. ISBN 978-0-471-47134-9.

[42] A. Jain e E. Y. Chang. Adaptive sampling for sensor networks. In *1st International Workshop on Data Management For Sensor Networks (DMSN'04)*, volume 72, pages 10–16, Toronto, Canada, August 2004. ACM.

[43] O. S. Junior, A. L. L. Aquino, e R. A. F. Mini. Um algoritmo de amostragem multivariada para redes de sensores sem fio. In *28st Brazilian Symposium on Computer Networks (SBRC'10)*, page 14, Gramado, RS, May 2010. SBC.

[44] O. S. Junior, A. L. L. Aquino, R. A. F. Mini, e C. M. S. Figueiredo. Multivariate reduction in wireless sensors networks. In *IEEE Symposium On Computers and Communications (ISCC'09)*, Sousse, Tunisia, July 2009. IEEE Computer Society.

[45] K. Kalpakis, K. Dasgupta, e P. Namjoshi. Maximum lifetime data gathering and aggregation in wireless sensor networks. In *2nd IEEE International Conference on Networking (ICN'02)*, pages 685–696, Atlanta, Georgia, USA, August 2002. IEEE Computer Society.

[46] H. Kim, J. Park, e G. Cho. Statistical data aggregation protocol based on data correlation in wireless sensor networks. In *International Symposium on Information Technology Convergence (ISITC'07)*, pages 130–134, Jeonju, Republic of Korea, November 2007. IEEE Computer Society.

[47] B. Krishnamachari, D. Estrin, e S. Wicker. The impact of data aggregation in wireless sensor networks. In *22nd IEEE International Conference on Distributed Computing Systems (ICDCS'02)*,

pages 575–578, Vienna, Austria, July 2002. IEEE Computer Society.

[48] W. J. Krzanowski. *Recent Advances in Descriptive Multivariate Analysis*, volume 2 of *Royal Statistical Society Lecture Notes Series*. Claredon Press, Oxford, July 1995. ISBN 978-0-19-852285-0.

[49] W. J. Krzanowski. *Recent advances in descriptive multivariate analysis*, volume 2. Oxford University Press, Oxford, USA, 1 edition, August 1995. ISBN 0198522851.

[50] J. Li e Y. Zhang. Interactive sensor network data retrieval and management using principal components analysis transform. *Smart Materials and Structures*, 15(11):1747–1757, December 2006.

[51] A. A. F. Loureiro, J. M. S. Nogueira, L. B. Ruiz, R. A. Mini, E. F. Nakamura, e C. M. S. Figueiredo. Wireless sensors networks (in portuguese). In *21st Brazilian Symposium on Computer Networks (SBRC'03)*, pages 179–226, Natal, RN, Brazil, May 2003. SBC.

[52] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, e T. He. Rap: A real-time communication architecture for large-scale wireless sensor networks. In *8th IEEE Real-Time Technology and Applications Symposium (RTAS'02)*, pages 55–66, San Jose, California, USA, September 2002. IEEE Computer Society.

[53] S. R. Madden, M. J. Franklin, J. M. Hellerstein, e W. Hong. Tinydb: An acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173, March 2005.

[54] G. Maia, D. Guidoni, A. L. L. Aquino, e A. A. F. Loureiro. Improving an over-the-air programming protocol for wireless sensor networks based on small world concepts. In *12th ACM International Conference on Modeling, Analysis and Simulation of Wireless*

and Mobile Systems (MSWIM'09), Tenerife, Canary Islands, September 2009. ACM Society.

[55] S. Mallat. *A Wavelet Tour of Signal Processing (Wavelet Analysis & Its Applications)*. Academic Press-Elsevier, Los Angeles, 1998.

[56] A. D. Marbini e L. E. Sacks. Adaptive sampling mechanisms in sensor networks. In *London Communications Symposium (LCS'03)*, pages 1–4, London, UK, September 2003. University College London.

[57] G. C. Menezes, A. Lins, R. da Silva Cabral, e F. G. Nakamura. Uma abordagem paralela para os problemas de cobertura e conectividade em redes de sensores sem fio. In *37th Brazilian Symposium on Operational Research (SBPO'05)*, pages 1–15, Gramado, RS, Brasil, September 2005. SOBRAPO.

[58] S. A. Mingoti. *Análise de dados através de métodos de estatística multivariada: uma abordagem aplicada*, chapter Anílise de Componentes Principais, pages 59–97. Editora UFMG, 2005. ISBN 85-7041-451-X.

[59] E. F. Nakamura, A. A. F. Loureiro, e A. C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3):9/1 – 9/55, April 2007.

[60] E. F. Nakamura, F. G. Nakamura, C. M. S. Figueredo, e A. A. F. Loureiro. Using information fusion to assist data dissemination in wireless sensor networks. *Telecommunication Systems*, 30(1–3): 237–254, November 2005.

[61] E. F. Nakamura, H. A. B. F. Oliveira, H. Ramos, L. A. Villas, A. L. L. Aquino, e A. A. F. Loureiro. A reactive role assignment for data routing in event-based wireless sensor networks. *Computer Networks*, 53:1980–1996, 2009.

[62] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[63] G. J. Pottie e W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[64] E. Reschenhofer. Generalization of the kolmogorov-smirnov test. *Computational Statistics & Data Analysis*, 24(4):422–441, June 1997.

[65] O. Roy e M. Vetterli. Dimensionality reduction for distributed estimation in the infinite dimensional regime. *IEEE Transactions on Information Theory*, 54(4):1655–1669, April 2008.

[66] E. M. Royer e C. K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[67] A. S. Ruela, A. L. L. Aquino, F. G. G. aes, e R. da S. Cabral. Redes de sensores sem fio modeladas como redes small world: uma heurística baseada em algoritmos genéticos. In *Simpósio Brasileiro de Pesquisa Operacional (XLI SBPO)*, Porto Seguro, BA, Brasil, Setembro 2009.

[68] A. S. Ruela, R. da S. Cabral, A. L. L. Aquino, e F. G. Guimarães. Evolutionary design of wireless sensor networks based on complex networks. In *Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'09)*, Melbourne, Australia, December 2009.

[69] A. S. Ruela, R. da S. Cabral, A. L. L. Aquino, e F. G. Guimarães. Memetic and evolutionary design of wireless sensor networks based on complex network characteristics. *International Journal of Natural Computing Research*, 1(2):33–53, April-June 2010.

[70] L. B. Ruiz, J. M. S. Nogueira, e A. A. Loureiro. Manna: A management architecture for wireless sensor networks. *IEEE Communications Magazine*, 41(2):116–225, February 2003.

[71] S. Santini e K. Romer. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *3rd International Conference on Networked Sensing Systems (INSS'06)*, pages 29–36, Chicago, IL, USA, 31 May – 2 June 2006. Transducer Research Foundation.

[72] C. Schurgers, V. Tsiatsis, S. Ganeriwal, e M. B. Srivastava. Topology management for sensor networks: Exploiting latency and density. In *3rd ACM International Symposium on Mobile Ad-Hoc Networking & Computing (MOBIHOC'02)*, pages 135–145, Lausanne, Switzerland, June 2002. ACM.

[73] S. Seo, J. Kang, e K. H. Ryu. Multivariate stream data reduction in sensor network applications. In *2nd International Symposium on Ubiquitous Intelligence and Smart Worlds (UISW'05)*, pages 198–207, Nagasaki, Japan, December 2005. Springer.

[74] S. Siegel e J. N. John Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill Humanities/Social Sciences/Languages, Columbus, OH, USA, 2 edition, January 1988. ISBN 0070573573.

[75] W. Song e X. Shaowei. Robust PCA based on neural networks. volume 1, pages 503–508, December 1997.

[76] N. Thomson. Understanding ANOVA the APL way. *ACM SIGAPL – APL Quote Quad*, 24(1):295–303, August 1993.

[77] S. Tilak, N. B. Abu-Ghazaleh, e W. Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, April 2002.

[78] N. Vlajic e D. Xia. Wireless sensor networks:to cluster or not to cluster? In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM'06)*, pages 258–268, Niagara-Falls, Buffalo-NY, June 2006. IEEE Computer Society.

[79] R. Willett, A. Martin, e R. Nowak. Backcasting: adaptive sampling for sensor networks. In *3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pages 124–133, Berkeley, California, USA, April 2004. ACM.

[80] H. Wu e Q. Luo. Supporting adaptive sampling in wireless sensor networks. *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 3442–3447, March 2007.

[81] L. Xu e A. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1):131–143, January 1995.

[82] J. Yick, B. Mukherjee, e D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52:2292–2330, 2008.

[83] Y. Yu, B. Krishnamachari, e V. K. Prasanna. Data gathering with tunable compression in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(2):276–287, February 2008.

[84] K. Yuen, B. Liang, e B. Li. A distributed framework for correlated data gathering in sensor networks. *IEEE Transactions on Vehicular Technology*, 57(1):578–593, January 2008.

[85] V. Zarzoso, P. Comon, e M. Kallel. How fast is fastICA? In *14th European Signal Processing Conference (EUSIPCO'06)*, Florence, Italy, September 2006.

[86] Y. Zhang, Y. Wang, D. Zhang, e C. Huang. Maximum-energy shortest path tree for data aggregation in wireless sensor

networks. In *3rd International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'07)*, pages 2779–2782, Shangai, China, September 2007. IEEE Computer Society.

[87] J. Zhao, R. Govindan, e D. Estrin. Residual energy scans for monitoring wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'02)*, pages 356–362, Orlando, Florida, USA, March 2002. IEEE Computer Society.

[88] Y. J. Zhao, R. Govindan, e D. Estrin. Computing aggregates for monitoring wireless sensor networks. In *1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03)*, pages 139–148, Anchorage, AK, USA, May 2003. IEEE Computer Society.

[89] R. Zheng e R. Barton. Toward optimal data aggregation in random wireless sensor networks. In *26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pages 249–257, Anchorage , Alaska , USA, May 2007. IEEE Computer Society.

[90] J. Zhu e S. Papavassiliou. A resource adaptive information gathering approach in sensor networks. In *IEEE Sarnoff Symposium on Advances in Wired and Wireless Communication (SARNOFF'04)*, pages 115–118, Princeton, NJ, USA, April 2004. IEEE Computer Society.

Índice