

**DETECÇÃO DE EVENTOS EM REDES DE  
SENSORES SEM FIO**

LETÍCIA DECKER DE SOUSA

DETECÇÃO DE EVENTOS EM REDES DE  
SENSORES SEM FIO

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

ORIENTADOR: ANTONIO ALFREDO FERREIRA LOUREIRO.

CO-ORIENTADOR: ALEJANDRO C. FRERY.

Belo Horizonte, Minas Gerais

Outubro de 2011

© 2011, Letícia Decker de Sousa.  
Todos os direitos reservados.

Decker de Sousa, Letícia  
S725d Detecção de eventos em redes de sensores sem fio /  
Letícia Decker de Sousa. — Belo Horizonte, Minas  
Gerais, 2011  
xv, 85 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais – Departamento de Ciência da  
Computação.

Orientador: Antonio Alfredo Ferreira Loureiro.  
Co-Orientador: Alejandro C. Frery.

1. Computação – Teses; Redes de Sensores Sem Fio –  
Teses. I. Orientador. II. Coorientador. III. Título.

CDU 519.6\*22(043)





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

## FOLHA DE APROVAÇÃO

Detecção de eventos em redes de sensores sem fio

**LETÍCIA DECKER DE SOUSA**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

  
PROF. ANTONIO ALFREDO FERREIRA LOUREIRO - Orientador  
Departamento de Ciência da Computação - UFMG

  
PROF. EDUARDO FREIRE NAKAMURA  
Departamento de Ciência da Computação - UFAM / FUCAPI

  
DR. LEONARDO BARBOSA E OLIVEIRA  
Instituto de Computação - Unicamp

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

Belo Horizonte, 27 de outubro de 2011.

*A Deus, aos meus amados Phillippe, Karin, Ronaldo, Alexander, Maiara, Evellyn, Prof. Alejandro e a todos meus amigos dessa jornada! =)*

# Agradecimentos

Agradeço a todos que, de uma forma ou outra, me ajudaram a concluir esse trabalho. Agradeço ao meu amado namorado, Phillippe Samer, que tantas vezes me incentivou e me ajudou a seguir em frente, mesmo quando as expectativas não eram muito animadoras. Pelas demoradas conversas sobre o trabalho e pelas inúmeras vezes que revisou meus textos e artigos. À minha família, minha mãe (Karin Decker de Sousa), meu pai (Ronaldo Carvalho de Sousa) e meu irmãos (Maiara e Alexander Decker de Sousa) que tantas vezes deixei em segundo lugar mesmo durante os fins de semana. Ao amor, compreensão e apoio que eles sempre me deram.

Aos amigos, companheiros de uma vida inteira, agradeço por ainda serem meus amigos, mesmo que eu fique muitos meses sem dar notícias ou pareça sempre escorregadia nos encontros casuais =P: Clarice Montanari, Milena Passini, Nivaldo César de Souza Júnior, Alessandra Vitor e tantos outros. Agradeço aos meus amigos de jornada: Glívia, Ismael, Ivre e Elisa, pelas conversas, estudos em grupo, carinho e apoio! Sem compartilhar as alegrias e frustrações do mestrado, o processo teria sido bem menos prazeroso!

Agradeço à calorosa acolhida dos amigos que fiz em Maceió durante o intercâmbio UFMG-UFAL realizado durante o mestrado: David, Paulinho, Vilker, Emanuelle, Manu, Thiago Véras e, em especial, minha lindeza Evellyn. Agradeço ao Prof. Alejandro Frery pelo carinho, pela atenção e pela orientação. Por me ensinar tantas coisas que vão além do mestrado e que vou levar para sempre no meu coração e na minha vida. Agradeço também ao apoio, às conversas, às revisões e às bricadeiras do Prof. André Lins.

Agradeço aos amigos do laboratório: Guilherme Maia, Leandro Vilas, Felipe Domingos, Heitor Ramos, Daniel Guidoni, Pedro Júnior, Marcos Nicolau, Bruno Lopes e, especialmente, Juliana Magalhães (minha amiga e ídola,rs), Eduardo Mucelli, Fernando e Thiago, pela companhia agradável durante o intercâmbio para o Centro de Pesquisa da Telecom Italia em Veneza, realizado no mestrado. E, finalmente, agradeço ao Prof. Loureiro por ter me convencido a fazer o mestrado! Obrigada a todos! =D

*“Não! Tentar não. Faça ou não faça. Tentativa não há.”*  
(Mestre Yoda)



# Abstract

Event detection is a fundamental problem in wireless sensor networks. Typical applications related to event detection include: medical and military developments, and environment monitoring. The goal of event detection is to identify when a collected data represents an event occurrence. The lack of uniformity on treating this problem does not allow the comparison among different proposals. This work presents a framework to detect events in scenarios with fault occurrence, measuring imprecision of the monitored characteristic performed by a non-ideal sensor device at a noisy environment. For this purpose, we extended the Diffuse framework to event detection, and evaluated it with a new proposed method based on Control Charts Theory. The Diffuse framework was initially proposed to detect sensor faults in a scenario with constant data streaming over a wireless sensor network using traffic metrics. The extended version of the Diffuse framework consists of five elements: (i) Measurement Processor, which is a module to filter the collected data and detect faults; (ii) Feature Processor, which transforms the collected data in a more significant entity; (iii) Event Processor, which is used to estimate the occurrence of a given event from a set of monitored characteristics; (iv) State Estimator, which infers the occurrence of events given the occurrence estimations of a neighborhood, and (v) Decision Maker, which chooses sending an event occurrence notification to the sink (in case it has been detected). Simulation results performed with the Sinalgo simulator showed that: (i) for 10% of sensor faults (i.e., when a sensor device measures a wrong data from the environment) our framework implementation presents approximately 96% of event detection hit rate; and (ii) for 30% of sensor faults, our framework implementation presents 75% of event detection hit rate. The contribution of this work is to provide an adapted version of the Diffuse framework that covers event detection methods based on estimation and inference theories using inaccurate collected data, providing a way to compare event detection implementations. Furthermore, we also provided a set of default models to compare event detection methods and two methods: new one based on Control Charts Theory and another based on Confidence Interval (inspired by a related work).

# Resumo

Detecção de eventos é um problema fundamental em redes de sensores sem fio. Algumas aplicações típicas são monitoramento de ambiente, aplicações médicas e militares. Durante o monitoramento, o objetivo é identificar quando os dados coletados identificam a ocorrência de um evento. A falta de uniformidade no tratamento desse problema dificulta a comparação entre diferentes propostas. Esse trabalho apresenta um arcabouço para detectar eventos em um cenário com ocorrências de falhas de sensores, cujas medidas possuem imprecisões provenientes de ruído do ambiente. Para tal, propomos a extensão do arcabouço *Diffuse* para detecção de eventos e o avaliamos com dois métodos. O *Diffuse* foi proposto inicialmente para detecção de falhas de sensores. A sua versão estendida é formada por cinco componentes: (i) Processador de Medida, que é um módulo de filtragem de dados coletados e detecção de falhas, (ii) Processador de Característica, usado para transformar a característica monitorada em outra que represente melhor o problema de interesse, (iii) Processador de Eventos, usado para estimar a ocorrência de um evento dado um conjunto de características; (iv) Estimador de Estado, que infere a ocorrência de eventos dadas as estimativas de ocorrências dos nós vizinhos e, (v) Tomador de Decisão, que opta pelo envio de uma notificação de ocorrência de evento para o nó sorvedouro, caso ele tenha sido detectado. Os resultados das simulações realizadas no *Sinalgo* mostram que: (i) para 10% de falhas de sensores (i.e., quando 10% dos sensores da rede medem um dado errado), nossa implementação do arcabouço apresenta aproximadamente 96% de taxa de acerto de detecção de eventos; e (ii) para 30% de falhas de sensores, nossa implementação do arcabouço apresenta 75% de taxa de acerto de detecção de eventos. A contribuição desse trabalho é fornecer a versão estendida do arcabouço *Diffuse*, que permita a implementação de métodos de detecção de eventos usando dados coletados com imprecisão, formalizando as comparações de métodos de detecção de eventos. Além disso, disponibilizamos uma modelagem-padrão para comparar métodos e verificamos a eficiência de nossa metodologia usando dois métodos: um novo baseado na Teoria de Gráfico de Controle e o outro inspirado em um trabalho relacionado.

# Lista de Figuras

4.1	Arcabouço <i>Diffuse</i> , traduzido de Nakamura (2007). . . . .	30
4.2	Arcabouço <i>Diffuse</i> estendido para o cenário de detecção de eventos. . . . .	31
4.3	Esquema de implementação do <i>Diffuse</i> estendido para o cenário de detecção de eventos. . . . .	33
4.4	Não há falhas no sensoriamento. . . . .	35
4.5	Há falhas no sensoriamento. . . . .	36
4.6	Não sabemos se houve falha durante o sensoriamento. . . . .	36
4.7	Fluxograma do CIM. . . . .	43
4.8	Probabilidade das medidas corretas estarem fora das linhas de controle. . . . .	45
4.9	Fluxograma de CCM. . . . .	47
4.10	Condições necessárias para recalculas as linhas de controle. . . . .	48
5.1	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CCM, variando-se a porcentagem de falhas de sensores. . . . .	57
5.2	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CCM, variando-se a quantidade de vizinhos e porcentagem de falhas de sensores. . . . .	59
5.3	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CCM, variando-se a quantidade de vizinhos e a porcentagem de falhas de sensores. . . . .	60
5.4	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CCM, variando-se a quantidade de vizinhos e porcentagem de falhas. . . . .	61
5.5	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CCM, variando-se a quantidade de vizinhos e porcentagem de falhas. . . . .	62
5.6	Taxa de detecção vs. porcentagem de vizinhos usados na inferência de eventos usando o CCM, variando-se a porcentagem de falhas de sensores. . . . .	63
5.7	Taxa de detecção vs. porcentagem de falhas de sensores usando o CCM. . . . .	64
5.8	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CIM, variando-se a porcentagem de falhas de sensores. . . . .	66

5.9	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CIM, variando-se a quantidade de vizinhos e a percentagem de falhas de sensores.	67
5.10	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CIM, variando-se a quantidade de vizinhos e a percentagem de falhas de sensores.	69
5.11	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CIM, variando-se a quantidade de vizinhos e a percentagem de falhas de sensores.	70
5.12	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CIM, variando-se a quantidade de vizinhos e a percentagem de falhas de sensores.	71
5.13	Taxa de detecção vs. desvio padrão da medida dos sensores usando o CIM, variando-se a quantidade de vizinhos e a percentagem de falhas de sensores.	72
5.14	Taxa de detecção vs. percentagem de vizinhos usados na inferência de eventos usando o CIM, variando-se a percentagem de falhas de sensores. . . . .	73
5.15	Taxa de detecção vs. percentagem de falhas de nós sensores usando o CIM.	74
5.16	Comparação das Taxa de detecção positivas dos métodos vs. percentagem de falhas de sensores. . . . .	75
5.17	Comparação das Taxa de detecção negativas dos métodos vs. percentagem de falhas de sensores. . . . .	76

# Lista de Tabelas

1.1	Resumo de Trabalhos Relacionados . . . . .	10
4.1	$\sigma_X$ é o desvio padrão do conjunto de dados e $n$ é o tamanho da amostra. . . . .	45
5.1	Modelos usados nas simulações. . . . .	51
5.2	Fatores usados nas simulações. . . . .	52
5.3	Parâmetros usados nas simulações. . . . .	53

# Sumário

Agradecimentos	vii
Abstract	ix
Resumo	x
Lista de Figuras	xi
Lista de Tabelas	xiii
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.2 Definição do Problema . . . . .	3
1.3 Trabalhos Relacionados . . . . .	4
1.3.1 Processador de Medida . . . . .	4
1.3.2 Processador de Característica . . . . .	5
1.3.3 Processador de Evento . . . . .	5
1.3.4 Estimador de Estado . . . . .	7
1.3.5 Tomador de Decisão . . . . .	9
1.3.6 Resumo . . . . .	9
1.4 Contribuições . . . . .	9
1.5 Estrutura do Documento . . . . .	9
<b>2 Evento</b>	<b>11</b>
2.1 Definições . . . . .	11
2.2 Características de Eventos . . . . .	14
2.2.1 Evento Simples e Composto . . . . .	14
2.2.2 Evento de Raio Fixo e Raio Variável . . . . .	14
2.2.3 Evento Estático, Móvel e Previsibilidade de Ocorrência . . . . .	15

2.2.4	Duração, Frequência e Suas Influências na RSSF . . . . .	15
2.3	Redes de Sensores Sem Fio Orientadas a Eventos . . . . .	16
2.3.1	Requisitos . . . . .	16
2.4	Considerações Finais . . . . .	17
<b>3</b>	<b>Modelagem do Problema</b>	<b>18</b>
3.1	Considerações Iniciais . . . . .	18
3.2	Problema Abordado . . . . .	20
3.3	Detalhamento dos Componentes do Problema . . . . .	21
3.3.1	Modelo de Medida de Interesse . . . . .	22
3.3.2	Modelo de Sensor . . . . .	23
3.3.3	Modelo de Falha . . . . .	24
3.3.4	Modelo de Evento . . . . .	25
3.3.5	Modelo de Detecção de Eventos . . . . .	26
3.4	Considerações Finais . . . . .	28
<b>4</b>	<b>Detecção de Falha e Eventos</b>	<b>29</b>
4.1	Arcabouço <i>Diffuse</i> . . . . .	29
4.2	Processo de Detecção . . . . .	32
4.2.1	Comparação de Intervalos . . . . .	34
4.2.2	Detecção de Falha . . . . .	38
4.2.3	Detecção do Evento . . . . .	39
4.3	Métodos Propostos . . . . .	41
4.3.1	Método do Intervalo de Confiança . . . . .	42
4.3.2	Método de Gráfico de Controle de Qualidade . . . . .	44
4.4	Considerações Finais . . . . .	48
<b>5</b>	<b>Simulação e Discussão dos Resultados</b>	<b>49</b>
5.1	Ferramenta de Simulação . . . . .	49
5.2	Descrição do Experimento . . . . .	50
5.3	Resultados e Análise . . . . .	54
5.3.1	Método Gráfico de Controle de Qualidade . . . . .	55
5.3.2	Método Intervalo de Confiança . . . . .	64
5.4	Considerações Finais . . . . .	77
<b>6</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>79</b>
	<b>Referências Bibliográficas</b>	<b>82</b>

# Capítulo 1

## Introdução

A interpretação do mundo que nos cerca é fornecida pela percepção das grandezas que o caracterizam. A fim de ampliar nossa capacidade de percepção do mundo, valemo-nos de instrumentos sensores. Eles são capazes de mensurar determinadas características físicas do meio e, a partir dessas observações, inferimos fatos.

Além do poder de sensoriar características físicas do ambiente, o sensor tem poder de processamento de dados e de comunicação com outros dispositivos. Tipicamente os dispositivos de sensoriamento trabalham de forma cooperativa e formam uma Rede de Sensores sem Fio (RSSF). Este tipo de rede é formada por um conjunto de dispositivos sensores, chamados nós fontes, que continuamente coletam dados do ambiente; por um ou mais nós especiais, chamados de nós sorvedouros, para os quais são encaminhados os dados coletados; e por um *gateway* que é uma interface de comunicação entre a rede de sensores e outras redes.

Pelas limitações de energia e processamento de dados impostas à rede de sensores, é necessário desenvolver técnicas eficientes sem afetar a eficácia das aplicações. Segundo Buratti et al. (2009), as aplicações para RSSFs são classificadas em: (i) detecção de eventos (DE) e (ii) estimação de processos espaciais (EPS). Em aplicações para DE, deseja-se identificar o instante e a localização de um determinado acontecimento como, por exemplo, focos de incêndios em uma floresta ou epicentros de terremotos. Em geral, estas aplicações possuem baixo processamento de sinal se comparadas à quantidade de medidas que podem ser feitas pelos aparelhos sensores, e fornecem ao nó sorvedouro uma notificação sobre a ocorrência de eventos quando eles forem detectados. Já em aplicações de EPS, o objetivo é monitorar uma dada característica física do ambiente no qual a RSSF está imersa, tal como umidade ou pressão atmosférica. Mesmo com objetivos finais diferentes, estas aplicações possuem soluções modeladas de maneira semelhante: coleta e tratamento contínuos de dados, diferindo no modo de entrega da



informação – quando se infere eventos em DE ou continuamente em EPS.

Dentre as aplicações que visam detectar eventos, uma possível classificação é por grupos de algoritmos. De acordo com Zoumboulakis & Roussos (2010), existem quatro grandes grupos de algoritmos para a detecção de evento complexo (DEC). Em aplicações de *DEC exata* ou *aproximada*, não se sabe inicialmente qual é o padrão de identificação do evento antes de se começar a usar o sistema. Já em *DEC não-paramétrica*, apesar de o padrão não precisar ser conhecido a priori, os nós sensores passam por um período de aprendizado, conhecendo os dados coletados. Os algoritmos de *DEC múltipla* propõem ser escaláveis para múltiplas detecções usando estruturas de dados eficientes para armazenamento de assinaturas de eventos. Finalmente, em *DEC probabilística*, os algoritmos são baseados na probabilidade da ocorrência de eventos. A escolha do tipo de algoritmo a ser usado depende do cenário que se deseja avaliar.

## 1.1 Motivação

Técnicas de detecção de eventos possuem uma grande aplicabilidade em RSSFs, além de serem facilmente adaptáveis ao contexto de monitoramento de ambiente e rastreamento de objetos. Ao considerarmos como eventos, mudanças de valor de uma dada característica física ou da posição de um objeto rastreado, as técnicas de detecção de eventos passam a ter um papel central no desenvolvimento de aplicações para RSSFs.

Alguns aspectos importantes na detecção de eventos são a eficácia e rapidez dos métodos, e a economia de recursos. Geralmente, as aplicações necessitam de uma resposta binária, em tempo real, à detecção de eventos. Neste cenário, temos o desafio de balancear tanto o processamento e a troca de mensagens quanto a eficácia.

A avaliação de algoritmos para RSSFs orientadas a eventos, assim como para outros tipos de rede, é feita, na maioria das vezes, através de simulações (Ai et al., 2009; Banerjee et al., 2008; Ding et al., 2005; Krishnamachari & Iyengar, 2004; Luo et al., 2006; Marco et al., 2007; Nakamura & Souza, 2010; Nittel, 2006; Sun & Coyle, 2010; Thuc & Insoo, 2011; Tseng et al., 2005; Wong & Sayeed, 2002; Yim & Choi, 2010). No entanto, há uma grande dificuldade em comparar os trabalhos devido à variabilidade das modelagens e dos cenários de detecção. Conforme é discutido no capítulo 3, elementos formadores do problema de detecção de eventos, tais como a medida realizada por um dispositivo sensor, o próprio dispositivo sensor, evento e sua detecção, são modelados de formas diferentes pelos estudos apresentados na literatura, tornando a comparação dos métodos muito custosa.

No melhor do nosso conhecimento, não existe nenhuma metodologia que forneça

um arcabouço e um conjunto de modelos para auxiliar na implementação de trabalhos que abordem o problema de detecção de eventos, facilitado a comparação entre os métodos.

## 1.2 Definição do Problema

Detecção de eventos, no contexto desse trabalho, é um tipo de monitoramento de ambiente que visa identificar a ocorrência de um acontecimento de interesse previamente caracterizado. A detecção é baseada na alta probabilidade dos sensores próximos fisicamente coletarem valores numéricos semelhantes.

Consideramos RSSFs em que o único fluxo de dados é feito quando se detecta um determinado evento. Essa rede possui dois estados que se alternam: “*evento*”, que acontece quando a RSSF detecta a ocorrência de um evento, e “*normal*”, caso contrário (Def. 3.2). O estado é estimado pelo valor da medida coletada pelo sensor, e cada dispositivo infere seu próprio estado atual. Portanto, dado um determinado evento simples e estático, do qual não se conhece inicialmente nenhuma propriedade além da característica física  $j$  que o define e os valores limite de  $j$  que o identificam, o problema central é estimar o atual estado da RSSF durante o tempo de vida  $T$  da rede (Def. 3.3).

Esse trabalho considera que há erro de precisão nas medidas feitas pelos dispositivos sensores, que há flutuações no valor da característica física monitorada e que há presença de ruído no ambiente. Consideramos, também, que os nós sensores podem falhar temporariamente durante o processo de medição.

Esse trabalho possui duas hipóteses. A primeira investiga se é possível tornar métodos de detecção diretamente comparáveis caso utilizem um mesmo arcabouço e uma mesma modelagem do problema para implementá-los. A segunda hipótese pesquisa sobre a possibilidade de desenvolver um método de detecção de eventos baseado na Teoria de Gráficos de Controle de Qualidade para RSSFs.

O objetivo geral desse trabalho é avançar o cenário atual de RSSFs Orientadas a Eventos (RSSF-OE) e incrementá-lo com a melhoria de técnicas de detecção baseadas em similaridade espaço-temporal.

Os outros objetivos são desenvolver uma metodologia que permita a implementação de técnicas de detecção de eventos que minimize os custos envolvidos durante a comparação de trabalhos relacionados e avaliar esta metodologia com técnicas propostas.

## 1.3 Trabalhos Relacionados

Esta seção discute trabalhos relacionados à detecção de eventos em RSSFs. Para tanto, apresenta-se resumidamente o arcabouço *Diffuse* estendido para cenários de detecção de eventos (Nakamura, 2007), que será detalhado no capítulo 4. Os componentes do arcabouço estendido são expostos no sentido de organizar as contribuições dos trabalhos analisados, categorizando-os por partes. O *Diffuse* estendido é formado por cinco componentes: (1) *Processador de Medida*, (2) *Processador de Característica*, (3) *Processador de Evento*, (4) *Estimador de Estado* e (5) *Tomador de Decisão*.

### 1.3.1 Processador de Medida

O Processador de Medida é o primeiro componente do *Diffuse* estendido, sendo utilizado para melhorar a qualidade da medida da característica de interesse  $j$  feita por um nó sensor  $i$ . O processamento da medida é necessário quando se deseja aproximar o processo de medição daquele que é realizado no mundo real, em que existem muitos fatores internos e externos aos dispositivos sensores que podem influenciar no valor observado. Isto é, esse processamento é necessário quando se trata de dados com anomalias, sejam elas causadas por imprecisão de instrumentos de medida, ruídos do ambiente ou ocorrências de falhas.

Durante essa etapa, o processador recebe a medida feita pelo nó sensor e a processa com o objetivo de melhorar a sua qualidade, sendo que uma maior qualidade nesse contexto significa obter uma medida mais próxima da medida real (*Def.* 3.5). Segundo Nakamura (2007), existem diversas técnicas que podem ser utilizadas durante essa fase do processo, como o cálculo de médias, máximos e mínimos. Técnicas de filtragem de sinais também são úteis durante essa etapa. Filtros de Partículas, de Média Móvel e de Kalman são exemplos de funções de estimação que ajudam a aumentar a confiabilidade da detecção de eventos.

É possível filtrar os dados considerando tanto a correlação espacial quanto a temporal, uma vez que se supõe que as características de interesse são bem comportadas em relação ao tempo entre amostragens. de Sousa et al. (2012a,b), Yim & Choi (2010), Nittel (2006) e Palpanas et al. (2003) usaram o filtro de média móvel baseados na correlação temporal dos dados coletados por um único sensor. Wong & Sayeed (2002) melhoraram a qualidade de sua coleta de dados usando modelos auto-regressivos (AR). Sun & Coyle (2010) utilizaram um filtro de mediana que se baseava na correlação espacial dos dados coletados por sensores próximos fisicamente. Ai et al. (2009) propuseram um algoritmo para filtrar os dados recebidos por um nó sensor durante o roteamento,

segundo o identificador do remetente do pacote para tratar injeção maliciosa de pacotes.

### 1.3.2 Processador de Característica

Assim como no *Diffuse* original, o processador de característica é usado para transformar a característica medida em uma entidade que represente melhor o problema de detecção de eventos. A maioria dos trabalhos analisados abordam problemas univariados e não usaram essa etapa do processo durante de detecção de eventos. A nossa abordagem também é univariada e a detecção de eventos é medida em termos da característica medida. Assim sendo, também não usamos esse componente do arcabouço.

### 1.3.3 Processador de Evento

O Processador de Evento é executado após o Processador de Medida, recebendo como entrada a medida tratada (e transformada caso seja usado o Processador de Característica) do nó sensor, e fornece como saída uma decisão binária (ou uma probabilidade, dependendo da abordagem) dessa medida representar a ocorrência de um evento. Assim, nesse componente são processadas técnicas de estimação para determinar a ocorrência de eventos.

Em Sun & Coyle (2010), a detecção de eventos era feita com a combinação de uma aproximação baseada em *Maximum a Posteriori* para decisões locais e globais em uma RSSF que utilizava comunicação multissalto, código de repetição de baixa complexidade e algoritmos de processamento de dados. Os nós sensores, que haviam detectado o evento, organizam-se em aglomerados (*clusters*) de maneira a formarem anéis concêntricos distanciados de um salto de comunicação. Todos os nós deste *cluster* enviam os dados para o nó agregador do *cluster* denominado *cluster head*, que sumariza os dados utilizando um algoritmo baseado em *Maximum a Posteriori*.

O trabalho apresentado por Yim & Choi (2010) utilizou um esquema de votação para identificar a ocorrência de um evento. Cada sensor tinha dois conjuntos de dados: um conjunto formado pela sua medida e aquelas dos nós vizinhos, e outro com estas medidas processadas por um filtro de média móvel. Usando um esquema de detecção por limiar (isto é, ocorre um evento em uma medição acima de um determinado valor), cada nó sensor estima a ocorrência de eventos localmente, utilizando os dois conjuntos de dados e um esquema de votação. Cada sensor tem seu nível de confiança dado por cada vizinho e outro calculado por si mesmo. A decisão final é a combinação dos resultados da estimação da ocorrência de eventos dados pelos dois diferentes grupos de

medidas, ponderados pelos níveis de confiança.

Bahrepour et al. (2010) trataram da detecção de evento em cenários de desastre, como fogo residencial, usando uma máquina de aprendizado distribuído para identificar o padrão de ocorrência de eventos. Cada nó sensor tem sua árvore de decisão baseada tanto em reputação quanto nos valores coletados pelos seus vizinhos em um esquema de votação.

Wong & Sayeed (2002) desenvolveram métodos para rastreamento de alvos que poderiam ser vistos como um tipo de detecção de eventos em que se deseja determinar quando e onde está um objeto. Seus métodos usam correlação espacial dada pelos  $k$ -vizinhos mais próximos ao nó sensor e um classificador baseado ou em Máxima Verossimilhança (*Maximum Likelihood*) usando um modelo de mistura de distribuição Gaussiana, ou em *Support Vector Machine*, que é um método de aprendizado baseado em reconhecimento de padrões.

Tseng et al. (2005) implementaram um sistema de visitante (*VS – Visitor System*), em que cada usuário se conecta e configura os seus eventos de interesse utilizando um sinal de celular ou *Bluetooth*. Ele tratou eventos como uma cadeia de regras de uma gramática. A saída do sistema é dada por uma expressão da gramática que define se o evento de interesse, que é pré-determinado pelo usuário, ocorreu. Os autores usam RSSFs e redes de telefonia celular para detectar e enviar mensagens de notificação de ocorrência de eventos. O *VS* envia notificações caso os eventos pré-configurados tenham ocorrido. A principal característica que diferencia este sistema de outros sistemas de mensagens é que ele possui um esquema de localização e de detecção de eventos.

Pawlowski et al. (2009) fizeram uma análise detalhada das variáveis envolvidas no monitoramento e controle de uma estufa. Para determinar se ocorreu um evento, subtrai-se o valor da medida atual pelo da anterior. Caso esta diferença exceda um determinado valor, então ocorreu o evento. As simulações deste sistema de monitoramento e atuação utilizaram dados reais.

Nittel (2006) utilizou um algoritmo baseado em correlação espacial para detectar o evento e identificar o seu limite físico. Este trabalho propôs um método para resolver estes problemas com duas variações: uma baseada no cálculo da média e a outra no da mediana dos valores coletados pelos vizinhos. Nesse trabalho, considera-se que ocorreu um evento se a média (ou mediana) calculada for maior que um certo limiar. Caso a medida realizada pelo sensor esteja dentro do intervalo de confiança dado pela média (ou mediana), o dispositivo sensor encontra-se na borda do evento. Com o objetivo de resolver os mesmos problemas, Ding et al. (2005) apresentaram um algoritmo em que cada nó sensor subtrai sua medida da mediana dos valores coletados pelos nós vizinhos obtendo um valor  $d$ . Os nós enviam seus valores de  $d$  para os vizinhos e calculam

a média  $\mu$  e o desvio padrão  $\sigma$  dos  $d$  recebidos para, então, calcular  $y$  ( $= \frac{d-\mu}{\sigma}$ ). A ocorrência de uma falha, que neste caso é o evento, acontece quando  $y$  for maior que um determinado limiar.

de Sousa et al. (2012a,b) apresentam dois diferentes métodos de detecção de eventos, um baseado na Teoria de Gráfico de Controle de Qualidade e outro baseado em intervalos de confiança. Ambos trabalhos são abordados nessa dissertação no capítulo 4.

### 1.3.4 Estimador de Estado

O Estimador de Estado da RSSF recebe um conjunto de estimação de evento da vizinhança e infere o estado do sensor dado um conjunto de estados possíveis {"evento", "normal"}, em que "evento" é quando a rede detecta a ocorrência de um evento e "normal", caso contrário.

Palpanas et al. (2003) modelaram as medidas dos nós sensores como uma função *kernel* e usaram um filtro de média móvel para verificar se ocorreu um valor anormal (considerado uma falha) no conjunto de dados formados pela sequência temporal coletada por cada nó sensor. Nesse trabalho, o evento era a ocorrência de uma falha e ela era determinada fazendo-se a composição dos resultados das detecções: de um nó para dois nós, de dois para três nós, e assim por diante.

Thuc & Insoo (2011) apresentaram uma técnica de detecção de eventos baseada em lógica difusa (*fuzzy logic*). O método de detecção é composto de três etapas: (i) a detecção local baseada na correlação espacial dos dados, (ii) a sumarização de dados realizada no nó agregador (*cluster head* ou *CH*) do conjunto de dados que detectou o evento (*cluster*), e (iii) fusão final centralizada que usa um esquema de credibilidade de *clusters* baseado em lógica difusa.

Krishnamachari & Iyengar (2004) propuseram um algoritmo baseado em votação para determinar a ocorrência de um evento usando decisão bayesiana. Eles apresentaram três algoritmos, todos envolvendo correlação espacial dos dados coletados pelos nós vizinhos. De acordo com os valores coletados pelos nós vizinhos, o algoritmo calcula a probabilidade bayesiana de o nó  $i$  ter detectado um valor correto e gera um número entre 0 e 1. No primeiro caso, gera-se um valor randômico entre 0 e 1, e se este valor for menor que a probabilidade calculada, assume-se que o nó  $i$  está correto. No segundo caso, se a probabilidade calculada do nó  $i$  for maior que um certo limiar, então o valor detectado está correto. No terceiro caso, se mais da metade dos vizinhos concordam com a decisão do nó  $i$ , então ele está certo. Luo et al. (2006) utilizaram um algoritmo baseado em inferência bayesiana com o critério de *Neyman-Pearson* para

detectar a ocorrência de um evento em um cenário tolerante a falhas. Baseado em um algoritmo de minimização de probabilidade de erros de detecção de eventos, cada nó sensor calcula o limiar de ocorrência de eventos, determina se sua medida representa um evento e envia a seus vizinhos sua decisão binária. Cada nó sensor soma os valores recebidos e, se esse valor for maior que  $k$  (valor previamente calculado pelo algoritmo de minimização), então o evento ocorreu.

Banerjee et al. (2008) trataram o problema de delimitação da região de ocorrência de evento. Os autores apresentaram um método de detecção baseado em um esquema de expressão polinomial em um cenário tolerante a falhas. O algoritmo é composto por duas fases: (i) o reconhecimento e (ii) a comunicação da ocorrência do evento e de sua área de abrangência. Primeiro, o método constrói uma árvore de agregação de dados, para a qual os sensores enviam sua localização, o valor coletado e os limites da região quadrada que mais se aproxima da área de ocorrência do evento. Os dados recebidos pela árvore são mapeados em um polinômio e esse processo continua até que os dados alcancem a raiz da árvore. Ai et al. (2009) apresentaram um sistema de detecção de eventos que é responsável por detectar a injeção de pacotes falsos, detectar eventos e roteá-los até o nó agregador do aglomerado. A árvore é construída para dar suporte à detecção de dados, transmitindo-os na forma de predicados. O evento de interesse é *temperatura > 150 & fumaça > 200 & umidade < 30*. Para identificar os pacotes falsos, filtram-se pacotes recebidos, comparando o identificador do remetente do pacote com uma lista de identificadores confiáveis gerada durante o processo de criação da árvore de roteamento. Se o pacote recebido não for de um nó confiável, descarta-o. Inicialmente, o agregador é responsável pela construção da árvore de detecção e, após gerá-la, manda mensagens de controle para os nós que pertencem a ela. Esse trabalho propôs um sistema de detecção de eventos chamado *Authentic Delay Bounded Event System (ADBES)*: identificar a injeção de pacotes falsos de origem maliciosa, detectar eventos e rotear pacotes para o nó sorvedouro. Esse esquema de detecção é baseado na confiança que se tem em cada nó e no sistema de segurança baseado em chaves. Apenas os nós que entram na árvore de roteamento podem encaminhar a detecção de eventos ao nó sorvedouro.

de Sousa et al. (2012a,b) usam um método de corte por limiar para melhorar a detecção de eventos realizada previamente durante a fase de processamento de evento. Essa técnica é abordada nessa dissertação no capítulo 4.

### 1.3.5 Tomador de Decisão

O Tomador de Decisão recebe como entrada o estado atual  $\Delta$  da rede e, de acordo com ele, escolhe qual é a melhor ação a ser tomada, que pode variar desde apenas reportar a ocorrência do evento ao nó sorvedouro até interferir na sua ocorrência com as ações de nós atuadores.

Dentre os trabalhos analisados, Ai et al. (2009), Tseng et al. (2005), de Sousa et al. (2012a,b) e Banerjee et al. (2008) apresentam esquemas de detecção que tratam desta última etapa. A decisão que eles tomam é reportar notificações sobre ocorrência de eventos caso elas aconteçam.

### 1.3.6 Resumo

A tabela 1.1 lista os trabalhos analisados e suas contribuições por cada componente do *Diffuse* estendido para detecção de eventos. Apresentando, também, os trabalhos realizados nessa dissertação (de Sousa et al., 2012a,b).

## 1.4 Contribuições

Esse trabalho propõe o arcabouço *Diffuse* estendido para detecção de eventos e uma modelagem que engloba os principais elementos do problema de detecção (tais como sensor, falha, medida, RSSF, evento e detecção) para tornar possível a implementação padronizada de cenários de detecção de eventos. Para avaliar o arcabouço, propomos um método de detecção de falhas e eventos baseado na Teoria de Gráficos de Controle de Qualidade e outro baseado em Intervalos de Confiança, e um método de inferência de eventos baseado em limiar.

## 1.5 Estrutura do Documento

O restante desse trabalho é organizado da seguinte maneira. O capítulo 2 apresenta uma discussão inicial sobre a definição e principais tipos de eventos. O capítulo 3 descreve os modelos utilizados para o sensor, a medida de interesse e as falhas previstas nas simulações da RSSF. São descritos ainda os modelos de evento e de sua detecção. O capítulo 4 compõe o cerne da dissertação, apresenta as técnicas de detecção utilizadas e o arcabouço *Diffuse* estendido. O capítulo 5 apresenta os experimentos realizadas com o simulador de algoritmos *Sinalgo*, utilizando os métodos propostos e comparando-os



Tabela 1.1: Resumo de Trabalhos Relacionados

Trabalho Relacionado	Processador de Medida	Processador de Evento	Estimador de Estado	Tomador de Decisão
de Sousa et al. (2012a,b)	Filtro de Média Móvel	Control Chart Method (CCM)	Control Chart Method (CCM)	Corte por Limiar
de Sousa et al. (2012a,b)	Filtro de Média Móvel	Confidence Interval Method (CIM)	Confidence Interval Method (CIM)	Corte por Limiar
Sun & Coyle (2010)	Filtro de Mediana	<i>Maximum a Posteriori</i> para decisões locais e globais	-	-
Yim & Choi (2010)	Filtro de Média Móvel	Esquema de votação com credibilidade + dois grupos de controle	-	-
Pawlowski et al. (2009)	-	Deteção temporal por limiar	-	-
Tseng et al. (2005)	-	Regras de gramática	-	Reporta notificação
Nittel (2006)	Filtro de Média Móvel	Correlação espacial usando <i>Média e Mediana</i>	-	-
Ding et al. (2005)	-	Cálculo da distância entre medida e <i>Mediana</i> dos vizinhos	-	-
Bahrepour et al. (2010)	-	Máquina de Aprendizado usando Árvore de Decisão e Reputação de nós	-	-
Wong & Sayeed (2002)	Modelos Auto-Regressivos	<i>k</i> -vizinhos mais próximos usando classificadores a base de <i>Máxima Verossimilhança</i> ou <i>Support Vector Machine</i>	-	-
Palpanas et al. (2003)	-	-	Filtro de Média Móvel composição de decisões locais	-
Thuc & Insoo (2011)	-	-	Deteção local + sumarização + <i>Lógica Difusa</i> com esquema de credibilidade	-
Krishnamachari & Iyengar (2004)	-	-	Decisão <i>bayesiana</i>	-
Luo et al. (2006)	-	-	Inferência <i>bayesiana</i> com o critério de <i>Neyman-Pearson</i>	-
Ai et al. (2009)	Filtro de ID	-	Árvore de Deteção	Reporta notificação
Banerjee et al. (2008)	-	-	Árvore de agregação com deteção polinomial	Reporta notificação

entre si. São descritos também os resultados das simulações e a análise dos dados. O capítulo 6 apresenta os trabalhos futuros e as conclusões desse trabalho.

# Capítulo 2

## Evento

Este capítulo discute o conceito de evento e suas características. A seção 2.1 apresenta e discute algumas definições de eventos. A seção 2.2 classifica e descreve diversos tipos de eventos. A seção 2.3 define o que é uma *Rede de Sensores sem Fio Orientada a Evento (RSSF-OE)*, levantando os requisitos dessa rede e a seção 2.4 sumariza os tópicos abordados.

### 2.1 Definições

“Evento” é um termo que possui diferentes significados, sendo amplamente difundido na vida cotidiana em diversas situações e contextos. Portanto, torna-se imprescindível definir qual é o significado que se aplica ao problema corrente. O leque de possibilidades abrange não apenas áreas distintas da Ciência da Computação como também de outras áreas do conhecimento, desde a Física até o senso comum e o vocabulário do dia-a-dia. De acordo com de Holanda Ferreira (1975), evento é um “*sucesso, acontecimento*”, ou, na abordagem estatística, é uma “*ocorrência em um fenômeno aleatório, de um membro de um determinado conjunto que se define a priori; acontecimento*”. Ainda em estatística, mas com maior formalismo, James (2008) afirma que evento é todo subconjunto  $A$  do espaço amostral  $\Omega$  do experimento, sendo  $\Omega$  o conjunto de resultados possíveis.

Já para Guralnik (1980), evento é definido como “*um determinado evento esportivo ou um item em um programa (salto com vara, salto em altura e outros eventos)*”.

Em uma outra abordagem, Int (1960) definiu evento como sendo “*uma ocorrência física envolvendo uma interação entre duas ou mais entidades físicas. Por exemplo, a colisão entre dois corpos; ou a emissão de um fóton por um átomo excitado como resultado de sua interação com um campo de radiação. Eventos são descritos como*

*ocorrências em posições e instantes de tempo específicos em um dado sistema de coordenadas, ou referencial. A teoria da relatividade especial trata tais problemas como a descrição do mesmo evento por observadores em dois referenciais inerciais equivalentes”.*

Esta abordagem é mais frequente nas ciências físicas, tal como Eletromagnetismo, Física Subatômica, Mecânica Relativística ou mesmo na Cosmologia. Mas é equivalente ao evento “acontecer algo”, adicionando-se as restrições de três coordenadas espaciais e a coordenada temporal para a identificação unívoca do evento.

Em uma outra definição, agora no contexto da Ciência da Computação, dada por Fragomeni (1987), um evento é um “*fato ou acontecimento de significação ou transcendência para uma tarefa ou sistema. Ex.: o término de uma operação assíncrona, tal como uma operação de entrada/saída*”.

Nesse sentido alguma mudança no estado de qualquer parte da memória pode ser considerado um evento, tal como a mudança no valor de uma variável previamente alocada. Ainda de acordo com o mesmo autor, “*um evento de um sistema, ou do meio ambiente (environment event), é uma mudança em uma ou mais propriedades estruturais do sistema, ou do meio ambiente, em um determinado período de tempo, isto é, uma mudança no estado estrutural*.”

Já Camarão (1989) afirmou que evento é “*o momento em que ocorre uma troca de estado específico. Normalmente um evento determina o término de uma operação de entrada e saída assíncrona*”. Essa definição permite construir sistemas como máquinas de estado, em que a troca de estado é dada pela ocorrência de um evento. Como sistemas de computação podem ser descritos como máquinas de estado, a ocorrência de evento surge com naturalidade nessa área de estudo.

Segundo Etzion & Niblett (2010), “*um evento é uma ocorrência dentro de um determinado sistema ou domínio; é algo que aconteceu, ou é considerado como ocorrido nesse domínio*”. Aplicando essa abordagem ao presente trabalho, o sistema seria a RSSF imersa no ambiente em que se deseja monitorar uma dada característica física, tal como a temperatura ou a umidade. Suponha que se deseja monitorar a temperatura de uma sala em um determinado lugar. O domínio dos valores que a temperatura pode variar é fornecido pelo histórico de temperatura do local em que esta sala se encontra e pela hora do dia. Qualquer valor detectado que não esteja dentro desse intervalo pode caracterizar, para o sistema, uma ocorrência de evento.

Em uma abordagem para RSSF, Banerjee et al. (2008) afirmaram que “*um evento pode ser definido como uma mudança excepcional nas características do ambiente, tais como temperatura, pressão, umidade, etc*”. Uma mudança excepcional é uma mudança atípica dos valores de uma determinada característica do ambiente e temos uma inter-

pretação parecida com a da definição anterior.

Em outra abordagem para RSSF, Ding et al. (2005) declararam que “*um evento, chamado de  $E$ , é um subconjunto de  $\mathcal{R}^2$  tal que as leituras de sensores em  $E$  são significativamente diferentes das dos sensores que não estão em  $E$ . Uma falha de um nó sensor pode ser vista como um tipo de evento especial em que contém apenas um ponto, i.e., o próprio sensor. O ponto  $x$  pertencente a  $\mathcal{R}^2$  é dito estar no limite de  $E$  se e somente se cada disco fechado centrado em  $x$  contém ambos pontos em  $E$  e fora dele.*”

Um evento pode ocorrer quando uma certa característica ou propriedade de uma determinada entidade alcança um dado valor. Um outro caso seria quando o evento é a ocorrência de um determinado acontecimento que não está vinculado a uma entidade, como por exemplo, se chove ou não. Nesse ponto de vista, pode-se considerar como eventos tanto ocorrências de fenômenos da natureza quanto valores assumidos por variáveis aleatórias, sendo elas vinculadas ou não a uma determinada entidade física.

Assim podemos ter dois tipos de eventos: (i) ‘*acontece de ser*’ e (ii) ‘*acontece algo*’. Em (i), monitora-se uma característica física vinculada a uma entidade, como a temperatura corporal de um paciente. Um evento acontece se esta temperatura alcançar um dado valor. Já em (ii), estamos monitorando características ambientais para determinar a ocorrência de fenômenos da natureza, tais como focos de incêndio em florestas. Neste caso, acontece um evento quando um conjunto de características ambientais combinadas representam a ocorrência do evento de interesse, tais como a concentração de monóxido de carbono no ar, umidade e temperatura, por exemplo. Desse modo, a maioria das aplicações em RSSFs pode ser interpretada como um tratamento de eventos: se temos uma entidade de interesse, monitoramos alguma de suas características que nos interessa. Se não a temos, monitoramos a ocorrência de um fenômeno, observando características ambientais.

Este trabalho considera evento como a combinação das duas últimas definições.

**Definição 2.1.** *Evento: é uma ocorrência ou a consideração da ocorrência de um determinado acontecimento dentro de um sistema ou domínio. O evento é identificado por uma mudança excepcional nas características do ambiente, tal como temperatura, pressão, umidade, etc, sendo que isto caracteriza a ocorrência de uma mudança de estado no sistema. Esse possui, pelo menos, dois estados que se intercalam: “ocorrência” e a “não ocorrência” de um evento em uma cadeia temporal.*

O objetivo de uma RSSF Orientada a Eventos é a captura de eventos no mundo real, isto é, a detecção de uma ocorrência externa ao sistema. Para detectar esse evento, monitora-se o ambiente de interesse, utilizando uma RSSF que, a partir de agora, será

denominada RSSF Orientada a Eventos (*Event-Driven Wireless Sensor Network*, Def. 3.1).

## 2.2 Características de Eventos

O capítulo 3 modela evento como uma tupla de características. Segundo a modelagem de evento adotada (vide Def. 3.14), evento é definido pelos seguintes itens: intervalo numérico que caracteriza a ocorrência do evento, área de ocorrência, mobilidade, previsibilidade, duração e frequência. Esta seção aborda cada característica que define um evento e fornece exemplos de aplicações para cada caso.

### 2.2.1 Evento Simples e Composto

Para detectar a ocorrência de um evento  $E$  é necessário definir o que o caracteriza. Por exemplo,  $E$  pode ser identificado quando a umidade relativa do ar de uma determinada sala alcançar um certo valor. Ou seja, precisamos saber a princípio qual é a característica física que é essencial para a detecção de  $E$  e como ela determina a sua ocorrência.

Assim sendo, existem dois grupos possíveis de eventos: (i) simples e (ii) composto. Em (i), apenas uma característica física  $j$  é necessária para a identificação de  $E$ , como no monitoramento da temperatura de uma estufa, em que a temperatura deve ficar entre certos limites que potencializam o crescimento das plantas. Temperaturas fora desse padrão caracterizam a ocorrência de um evento. Já em (ii), para determinar a sua ocorrência é preciso combinar diferentes características coletadas do ambiente, como na detecção de focos de incêndio. Nesse caso, a ocorrência pode ser determinada em conjunto pela concentração de monóxido de carbono no ar, pela umidade relativa do ar e pela temperatura.

Nesse trabalho, detectamos eventos simples. Mas nossa abordagem pode ser facilmente estendida para detecção de eventos compostos.

### 2.2.2 Evento de Raio Fixo e Raio Variável

O evento  $E$  pode ter raio fixo, isto é, sua área de influência não se modifica com o passar do tempo. Esse comportamento pode ser observado em rastreamento de alvos, monitoramentos de animais e detecção de pessoas em um ambiente, por exemplo. Nesses casos, o evento  $E$  é a detecção de um corpo rígido de tamanho determinado durante todo o tempo de monitoração.

A outra possibilidade é o evento de raio variável. Um exemplo seria a detecção de focos de incêndio ou o monitoramento de estruturas como pontes e prédios. O evento seria a detecção da área afetada pelo fogo e da área que sofreu algum tipo de estresse, respectivamente.

### 2.2.3 Evento Estático, Móvel e Previsibilidade de Ocorrência

Outro fator importante na caracterização de um evento é sua mobilidade. Eventos estáticos são aqueles que não se movem durante seu tempo de vida. O local de aparecimento do evento e de sua extinção é o mesmo e ele permanece no mesmo lugar durante todo o período de observação. Um exemplo disto é o monitoramento do uso de lâmpadas em um prédio. Nesse cenário, o evento é estático, pois consideramos lâmpadas fixadas em uma estrutura e seus comportamentos podem ser previstos, isto é, seguem um comportamento sazonal que depende da hora do dia e da época do ano.

Quando se monitora um evento móvel, deve-se considerar, também, as características do movimento do evento  $E$ . A trajetória de um evento pode ser descrita por uma função temporal ou uma distribuição de probabilidade, ou mesmo, ter um comportamento não-determinístico. Uma aplicação em que se monitora a queda livre de algum objeto é um exemplo de trajetória que segue uma função temporal e o monitoramento de partículas suspensas em um líquido, conhecida como *movimento browniano*, é uma aplicação em que temos um comportamento não-determinístico.

Normalmente, a ocorrência de eventos possui um fator de aleatoriedade envolvido, ou seja, onde e quando o evento ocorre não é pré-determinado. Mas mesmo assim, alguns eventos possuem um comportamento conhecido. Temos exemplos clássicos de eventos que seguem determinados tipos de distribuições de probabilidades tal como a ocorrência de eventos raros, que é descrita por uma distribuição de Poisson, ou o tempo de vida de equipamentos, que é dado por uma distribuição Exponencial.

### 2.2.4 Duração, Frequência e Suas Influências na RSSF

A duração do evento é outro ponto-chave na estratégia de detecção. Quando sua duração ultrapassa ou mesmo representa parte substancial do tempo de vida da rede, o monitoramento é prejudicado pelo esvaimento dos recursos da RSSF. Por esse motivo, o monitoramento de eventos usando uma RSSF-OE apenas faz sentido quando o evento possui baixa frequência e curta duração. Aplicações de detecção de eventos de alta frequência são mais comuns em redes de emergência, em que há necessidade de detectar um número máximo de eventos durante um curto intervalo de tempo, como

durante a busca por sobreviventes após deslizamentos de terra.

Esse estudo dedica-se a cenários em que se pretende prolongar o tempo de vida da RSSF, ou seja, na detecção de eventos de baixa frequência e curta duração.

## 2.3 Redes de Sensores Sem Fio Orientadas a Eventos

Uma RSSF Orientada a Eventos (RSSF-OE) é uma RSSF que se propõe identificar a ocorrência de eventos de uma dada aplicação. No momento da detecção do evento  $E$ , a RSSF-OE pode notificar urgentemente o nó sorvedouro ou encaminhar a ele os dados a uma taxa fixa. A notificação urgente de  $E$  existe quando há necessidade de uma ação de emergência, como em detecção de incêndios. Nesse cenário, o tráfego de dados na rede pode ocorrer apenas quando um evento é detectado. O outro caso é aquele em que estamos interessados no estudo de um determinado evento. Nesse cenário, as mensagens de notificação não são urgentes, sendo entregues a uma taxa fixa.

O comportamento de eventos em diferentes aplicações possui uma natureza estocástica. Isto acontece porque existem muitos fatores envolvidos durante todo processo e muitos deles também possuem um comportamento estocástico. A necessidade de monitoramento existe justamente por causa dessa característica dos sistemas complexos: o não-determinismo rege muitos dos eventos de interesse.

### 2.3.1 Requisitos

Cada um dos dois tipos de RSSF-OE, descritos acima, possui exigências de requisitos diferentes para o seu funcionamento. Uma RSSF-OE em que se deseja acionar uma ação sempre que um evento é detectado caracteriza-se pela necessidade de baixa taxa de erros de detecção e rapidez na entrega de notificações. Essas necessidades surgem porque os efeitos, intensificadores ou paliativos, são necessidades de tempo real. O envio de mensagens de notificação ao nó sorvedouro é feito sob demanda, a uma taxa variável no tempo.

O mesmo não ocorre em RSSF-OE que visam estudar as características de um determinado evento. Esse tipo de rede não precisa informar ao nó sorvedouro a ocorrência do evento no instante em que ele ocorre. Ela trabalha a uma taxa fixa de envio de mensagens ao nó sorvedouro e não existe a necessidade de rapidez nem de garantia de entrega.

Esse trabalho estuda o comportamento da RSSF orientada a eventos com as seguintes especificações:

- Existe um único nó sorvedouro para o qual todos os demais nós sensores encaminham eventualmente a ocorrência do evento de interesse.
- O nó sorvedouro e os demais nós sensores são estáticos.
- A RSSF-OE não possui uma comunicação contínua de dados ao nó sorvedouro, apesar de os dispositivos sensores estarem continuamente coletando dados. Os dados coletados somente são enviados ao nó sorvedouro quando há uma detecção positiva do evento.
- A RSSF-OE é depositada no formato de uma grade regular.
- Cada nó, que não os da borda, possui tipicamente oito vizinhos e eles estão distantes até  $t$  unidades de medida.
- O raio de sensoriamento e de comunicação de um nó sensor é  $\sqrt{2t^2}$  unidades de medida.

## 2.4 Considerações Finais

Esse capítulo apresentou uma discussão sobre o termo *evento* em toda sua generalidade e enuncia a definição adotada nesse trabalho. Outra discussão feita é sobre possíveis tipos de evento, tendo em vista a definição de evento utilizada. Finalmente, expomos uma breve argumentação sobre os requisitos de uma rede de sensores que se propõe a detectar eventos.

O próximo capítulo enuncia o escopo do trabalho e o problema que será abordado, detalhando os modelos envolvidos para a proposição de uma possível solução para esse problema. Os modelos propostos em conjunto com o arcabouço *Diffuse* estendido para detecção de eventos (abordado no capítulo 4) fornecem uma metodologia sistemática de implementação de cenários para o problema tratado.



# Capítulo 3

## Modelagem do Problema

Esse capítulo apresenta a modelagem do problema de detecção de eventos usada nessa dissertação. A seção 3.1 faz um levantamento bibliográfico de modelagens em trabalhos relacionados. A seção 3.2 descreve o problema tratado. A seção 3.3 apresenta a modelagem considerada do problema. A seção 3.4 expõe as considerações finais, resumindo o assunto tratado no capítulo.

### 3.1 Considerações Iniciais

O principal desafio na detecção de eventos é diminuir a distância entre a detecção e a realidade, minimizando os casos de falsos positivos e negativos em cenários em que há, dentre outros fatores, falhas de medidas de nós sensores e ruídos. A modelagem feita nesse trabalho visa uma maior aproximação entre simulação e cenário real ao considerar os fatores citados. Além disso, objetiva definir os elementos envolvidos na modelagem do problema de detecção.

O processo de detecção de eventos, em geral, possui as etapas de coleta, processamento e decisão. Os nós sensores coletam informações sobre as características de interesse do ambiente, processam esses dados para, então, inferirem sobre a ocorrência de eventos. Durante esse processo, um fator importante é a modelagem das medidas feitas pelos nós sensores. Em Thuc & Insoo (2011), uma “medida” foi modelada como um sinal que decai com o aumento da distância ao ponto central da ocorrência do evento, adicionado de um ruído Gaussiano. Luo et al. (2006) modelaram a medida de maneira similar. Nesse trabalho, a medida de um nó sensor é  $\pm d$  somado a um ruído Gaussiano, sendo que  $d$  é um dos fatores do experimento. Em Nittel (2006), a medida é modelada como uma função que depende da localização do nó sensor somada a um ruído branco Gaussiano. Já em outros trabalhos, ela é modelada de maneira bem mais

simples. Em Krishnamachari & Iyengar (2004) e em Yim & Choi (2010), a medida é tratada como uma variável aleatória binária; em Sun & Coyle (2010), como uma variável aleatória de Bernoulli; e em Ding et al. (2005), como uma variável aleatória Gaussiana. Outro exemplo de modelagem é em Banerjee et al. (2008). Nesse trabalho, a característica de interesse é a temperatura e a sua medida é modelada como sendo uma função de distribuição Gaussiana truncada entre dois valores limites. Já em outros trabalhos, o evento não precisou ser modelado, já que foram utilizados dados reais (Wong & Sayeed, 2002; Zoumboulakis & Roussos, 2010).

Outro fator importante na modelagem do cenário de interesse é a ocorrência de falhas de nós sensores. Para Ding et al. (2005), a presença de falhas constitui um desafio para a detecção de eventos e de seus limites físicos. As falhas podem ser proveniente de falhas de *hardware*, ataques de segurança ou distúrbios ambientais. Palpanas et al. (2003) definiram falha como um *outlier*, sendo “*uma observação que parece desviar-se significativamente de outros membros da amostra em que ocorre*”. Novamente segundo Ding et al. (2005), falhas de nós sensores são geograficamente independentes. No entanto, isso já não acontece quando ocorrem eventos, sendo esse um fator diferencial entre estes dois tipos de ocorrências. Por essa razão, há um interesse dos nós sensores em compartilhar suas informações com os demais com a finalidade de aumentar a confiabilidade durante o processo de detecção (Gungor & Akan, 2007). A maioria dos métodos de detecção de eventos em RSSFs assume que existe uma correlação espacial entre os dados coletados pelos dispositivos (Ai et al., 2009; Bahrepour et al., 2010; Ding et al., 2005; Krishnamachari & Iyengar, 2004; Luo et al., 2006; Nittel, 2006; Palpanas et al., 2003; Sun & Coyle, 2010; Thuc & Insoo, 2011; Wong & Sayeed, 2002; Yim & Choi, 2010; Zoumboulakis & Roussos, 2010) e, baseado nisso, utiliza técnicas de sumarização de dados ou métodos para estimar ou inferir a ocorrência de eventos.

Existem vários tipos de falhas que podem ocorrer durante a detecção de um evento. Sun & Coyle (2010) e Pawlowski et al. (2009) consideram ocorrências de erros de medida dos sensores e de erros de comunicação (*channel noise, fading, transceiver errors*). Tseng et al. (2005) consideram a ocorrência de erros de comunicação, e Krishnamachari & Iyengar (2004) consideram erros de medidas, ambos individualmente. Já Ai et al. (2009) consideram a injeção maliciosa de pacotes falsos durante o processo de fusão de dados para entregas de notificações de ocorrência de eventos. Yim & Choi (2010) exploram amplamente o cenário de falhas. Segundo os autores, o desempenho da detecção diminui quando aumenta a probabilidade de falhas. Assim sendo, Yim & Choi (2010) tratam a ocorrência de três tipos de falhas: (1) permanentes ou catastróficas, (2) temporárias e (3) intermitentes. Em (1), assumiu-se que a probabilidade de detectar um evento é de 50%, consistentemente, independente da região em que se encontra

o nó sensor. Já em (2), elas possuíam independência espaço-temporal. E, por fim, em (3) a falha é caracterizada pela ocorrência periódica, sendo de ajuda para estimar a capacidade de sistemas adaptarem-se a um esquema de falhas. Além disso, o ruído é outro fator prejudicial à qualidade de detecção de eventos, sendo o ruído Gaussiano o mais comumente considerado (Krishnamachari & Iyengar, 2004; Luo et al., 2006; Thuc & Insoo, 2011).

Os trabalhos podem utilizar dados coletados em ambientes reais ou fazer simulações de cenários para testar seus métodos de detecção. Bahrepour et al. (2010), Pawlowski et al. (2009), Kruger & Grosse (2009) e Zoumboulakis & Roussos (2010) usam dados reais para avaliar seus algoritmos de detecção. Uma aplicação prática recorrente tem sido o monitoramento de estruturas usando uma RSSF (Jeong et al., 2010; Kruger & Grosse, 2009). Nesse caso, o evento a ser detectado são danos causados às estruturas físicas por agentes externos, como o tempo, umidade ou mesmo tráfego. No entanto, a maioria dos trabalhos em RSSFs orientadas a eventos são simulados (Ai et al., 2009; Banerjee et al., 2008; Ding et al., 2005; Krishnamachari & Iyengar, 2004; Luo et al., 2006; Marco et al., 2007; Nakamura & Souza, 2010; Nittel, 2006; Sun & Coyle, 2010; Thuc & Insoo, 2011; Tseng et al., 2005; Wong & Sayeed, 2002; Yim & Choi, 2010), e a comparação entre esses trabalhos é muito custosa por causa da diversidade de modelos utilizados para a implementação dos elementos básicos do problema de detecção.

O problema abordado nesse capítulo é a descrição dos modelos dos fatores envolvidos na detecção de um evento simples utilizando RSSFs. A seção 3.2 apresenta o escopo e o problema abordado nesse trabalho. A seção 3.3 discute cada fator envolvido no problema. A seção 3.4 apresenta as considerações finais sobre os desafios envolvidos.

## 3.2 Problema Abordado

Essa seção enuncia o escopo e o problema que serão abordados nesse trabalho. Primeiramente, definimos o que chamaremos de RSSF-OE a partir do que já foi visto nos capítulos anteriores.

**Definição 3.1.** *RSSF-OE: Toda RSSF que se destina a detectar ocorrências de eventos é uma RSSF-OE. Assumindo uma topologia de grade 2D, representa-se a rede como um grafo  $G(V, A)$ , com as seguintes propriedades:*

- $V$  é um conjunto de  $n$  vértices, sendo que cada vértice representa um nó sensor. Um desses nós é o sorvedouro, para o qual se encaminham todas as notificação de ocorrência de eventos durante o tempo de vida  $T$  da RSSF-OE. Os outros

$n - 1$  nós pertencentes a  $V$  são os nós fontes, que se dedicam a monitorar a característica física  $j$  que identifica o evento de interesse  $E$ . Os nós sensores têm o tempo de vida útil menor ou igual a  $T$ .

- $A$  é um conjunto de arestas definidas por pares de nós sensores pertencentes a  $V$  que representam conexões ativas entre eles.
- Assume-se ainda que nenhum nó mantém um enlace de comunicação consigo mesmo, que existe no máximo um enlace entre um dado par de vértices e que os enlaces de comunicação são simétricos, ou seja, se o nó  $X$  alcança  $Y$ ,  $Y$  alcança  $X$ .

**Definição 3.2.** *Escopo do Problema:* Considere uma RSSF-OE, em que apenas há um fluxo de dados quando se detecta um determinado evento. A rede possui dois estados que se intercambiam: “evento” e “normal”. O estado “evento” acontece quando a RSSF-OE detecta a ocorrência de um evento, e o estado “normal”, caso contrário.

**Definição 3.3.** *Definição do Problema:* Dado um determinado evento simples e estático, do qual não se sabe nenhuma propriedade além da característica física  $j$  que o define e os valores limites de  $j$  que o identificam, o problema consiste em distinguir quando ocorrem as mudanças de estados da RSSF-OE durante  $T$ .

Esse problema possui uma grande aplicabilidade. Problemas baseados em monitoramento de ambientes e rastreamento de alvos podem ser modelados como sistemas de detecção de evento e, assim, podemos utilizar soluções similares para diferentes aplicações. O monitoramento de ambiente pode visar tanto estudar certas características físicas de um determinado local quanto identificar padrões nessas características e vê-los como eventos. O rastreamento de alvos pode ser modelado também como um tipo de monitoramento de ambiente que visa identificar qual a localização de um determinado alvo em um dado instante. Assim sendo, soluções propostas para detecção de eventos podem ser adaptadas para as principais classe de aplicações de RSSF.

### 3.3 Detalhamento dos Componentes do Problema

Essa seção apresenta os modelos usados nesse trabalho, assim como as premissas consideradas. Detalham-se os modelos utilizados para medir as características de interesse, nó sensor, evento e sua detecção.

### 3.3.1 Modelo de Medida de Interesse

Como apresentamos no capítulo 1, o objetivo desse trabalho é o desenvolver técnicas que visam detectar a ocorrência de eventos simples utilizando RSSFs, isto é, identificar quando as medidas da característica física  $j$  representam a ocorrência de um evento. Essa característica é monitorada por um dispositivo eletrônico que possui um elemento sensor para essa característica. Essa medida é composta por três componentes: (1) a medida real da característica de interesse, (2) erro de medida do dispositivo sensor e (3) ruído provido pelo ambiente. Os componentes são modelados como variáveis aleatórias, assim como a medida de interesse. Essa abordagem é conhecida como modelo de *Grubbs* (Grubbs, 1948, 1973) e é descrito a seguir.

**Definição 3.4.** *Modelo de Grubbs: De acordo com Lopes (2004), Grubbs (1948, 1973) supõe que um sistema com  $p$  instrumentos de medidas, sendo que  $p \geq 3$ , é usado para medir a característica de interesse  $j$  de um grupo de características de tamanho  $n$ . Sejam*

- $x_j$  o verdadeiro valor da  $j$ -ésima característica de interesse e
- $y_{ij}$  a medida da  $j$ -ésima característica de interesse feita pelo instrumento  $i$ , sendo que  $i = 1 \dots p$  e  $j = 1 \dots n$ .

Ainda segundo Lopes (2004), Grubbs (1948, 1973) propõem o seguinte modelo:

$$y_{ij} = x_j + \alpha_i + \epsilon_{ij} \quad (3.1)$$

sendo que  $\alpha_i$  é o vício aditivo do instrumento  $i$ , ou seja, o erro provocado pela precisão limitada do instrumento  $i$  durante o processo de medição e  $\epsilon_{ij}$  são erros aleatórios adicionados ao valor mensurado que dependem do instrumento  $i$ , da característica  $j$  e do ambiente monitorado. Nesse trabalho, supomos que os erros são independentes uns dos outros e descritos por uma distribuição Uniforme entre 0 e 1, por simplificação da implementação do filtro de sinal. A escolha do tipo de filtro a ser utilizado está associada ao tipo de ruído de interesse do estudo. Em casos em que desejamos estudar ruído Gaussiano supõe-se que

$$E(\epsilon_{ij}) = 0, Var(\epsilon_{ij}) = \phi_i, \quad (3.2)$$

sendo que  $i = 1 \dots p$  e  $j = 1 \dots n$ .

Para definir a medida de um nó sensor, é necessário definir a medida real da característica de interesse  $x_j$ , erro de medida do dispositivo sensor  $\alpha_i$  e ruído  $\epsilon_{ij}$ .

**Definição 3.5.** *Medida Real da Característica de Interesse  $x_j$ : Dado um dispositivo sensor ideal, a medida da característica  $j$  realizada por esse sensor é a medida considerada verdadeira. Supõe-se que a característica de interesse é uma variável aleatória.*

**Definição 3.6.** *Erro de Medida do Dispositivo Sensor  $\alpha_i$ : Dado um elemento sensor qualquer e supondo ausência de ruído no ambiente, o erro proveniente do equipamento de medida é definido como sendo a medida coletada pelo dispositivo sensor menos a medida real  $x_j$ . O erro de medida do sensor é uma variável aleatória.*

**Definição 3.7.** *Ruído  $\epsilon_{ij}$ : Consideram-se ruídos as flutuações constantes nas vibrações de um sistema mecânico ou eletrônico, limitando a sensibilidade de tais equipamentos (Kannan, 1979). O ruído é uma variável aleatória.*

Com isso, podemos definir o que consideraremos como sendo a medida do nó sensor.

**Definição 3.8.** *Medida do Nó Sensor  $y_{ij}$ : Considera-se que a medida feita por um nó sensor é a somatória dos componentes da medida, isto é, a medida real da característica de interesse  $x_j$  (Def. 3.5), o erro da medida do sensor  $\alpha_i$  (Def. 3.6) e o ruído branco randômico  $\epsilon_{ij}$  (Def. 3.7). Como todos os componentes da medida são modelados como variáveis aleatórias, a medida do nó sensor também o será.*

Existem vários tipos de ruídos. Dentre eles, podemos citar *shot noise* que é obtido em tubos a vácuo devido a emissões randômicas de elétrons por cátodos aquecidos. Outro exemplo é o ruído térmico devido a agitações térmicas de elétrons livres em resistores, ocorrendo principalmente em rede elétrica e sendo causado pela variação na voltagem nas extremidades dos resistores (Kannan, 1979). Por motivo de simplificação, escolhemos o ruído branco randômico modelado por uma *Distribuição Uniforme* entre 0 e 1.

### 3.3.2 Modelo de Sensor

Suponha que seja possível construir um dispositivo sensor ideal, ou seja, um elemento sensor cuja acurácia seja infinita. Mesmo assim, teríamos um limite físico de precisão de medida dado pelo Princípio de Incerteza de *Heisenberg* igual a  $h/4\pi$ , sendo  $h$  a constante de *Planck* (Eisberg & Resnick, 1923). Essa constante possui um valor muito pequeno e apenas é perceptível em experimentos que envolvem partículas atômicas. No entanto, isso nos fornece um argumento sobre a imprecisão inerente ao processo de medição e a importância de desenvolver modelos que se aproximem à realidade de cenários de coleta de dados.

Tal suposição a respeito de instrumentos de medida ideais não é válida nem para os dispositivos eletrônicos nem mecânicos da atualidade e, nesse sentido, assumimos que existe um erro de medida associado ao instrumento de medida contido no dispositivo sensor. Outro aspecto relevante é que, segundo Gaura & Newman (2006), há uma tendência na fabricação de nós sensores que visa a minimização do custo e, espera-se, portanto, uma menor precisão dos equipamentos. Em geral, as medidas são feitas em condições não-ideais de coleta: sensores com erros de medida associados, efeitos de interferência e ambiente com diversas fontes de ruído.

Assumimos nesse estudo a seguinte definição para o dispositivo sensor:

**Definição 3.9.** *Dispositivo Sensor: Elemento básico da RSSF que visa monitorar, com relação a uma determinada característica física, o ambiente no qual ele está imerso. É um dispositivo eletrônico não-ideal  $i$  e, como tal, suas medidas possuem um erro associado  $\alpha_i$ . Ele também está sujeito a falhas de origem desconhecida (Def. 3.13) de carácter temporário.*

**Definição 3.10.** *Tempo de Vida Útil de um Dispositivo Sensor: É o período de tempo em que o nó sensor realiza medidas que se distanciam da medida real  $x_j$  por, no máximo, uma vez o erro de precisão conhecido  $\alpha_i$  adicionado da variância do ruído de fundo  $\epsilon_{ij}$ .*

### 3.3.3 Modelo de Falha

Suponha que a característica de interesse  $j$  é descrita de forma bem comportada no espaço e no tempo, ou seja, coletas feitas por sensores próximos fisicamente ou em sequência por um mesmo sensor  $i$  possuem medidas com valores similares. Suponha ainda que existam ruídos no ambiente que podem distanciar uma medida do conjunto de dados coletados.

O desafio envolvido nesse cenário é conseguir distinguir entre ocorrência de anomalias e eventos, já que ambos diferem do conjunto de dados em algumas situações. No caso de eventos, a correlação entre os dados é menor durante seu início e término, e também nas bordas de sua área de influência. Já quanto à ocorrência de anomalias, podem existir muitas causas, tais como falhas de nós sensores, ataques maliciosos e invasão, ou fenômenos não-usuais na área de detecção (Rajasegarar et al., 2008).

Esse trabalho considera apenas falha de nós sensores, independente da sua origem, e assume-se que ela é temporária, isto é, o dispositivo sensor torna-se completamente não-funcional por um período de tempo e depois retorna a fazer medidas corretas.

Consideramos também que o ambiente insere ruídos nas leituras das características de interesse de modo contínuo.

Definimos, portanto, anomalia como se segue.

**Definição 3.11.** *Anomalia: Em um grupo de dados, é uma observação que é inconsistente com o restante dos dados (Rajasegarar et al., 2008).*

Assumimos que detecção de anomalia segue a seguinte premissa.

**Definição 3.12.** *Detecção de Anomalia: A detecção de um comportamento anormal sem saber, a priori, como se parece um comportamento anormal é a detecção de uma anomalia no conjunto de dados (Rajasegarar et al., 2008).*

Assim sendo, podemos definir a falha na medida no contexto desse trabalho.

**Definição 3.13.** *Falha de Medida: Dado um dispositivo sensor, destinado a monitorar uma determinada característica física  $j$  do ambiente, ele apresentará uma falha de medida quando, ao mensurar  $j$ , obtiver um valor inconsistente com o conjunto de dados já observados pelo nó sensor ou pelos seus vizinhos. A falha acontece devido a erros provenientes do dispositivo sensor e não por causas oriundas do meio (ruído) ou limitações físicas do equipamento. A falha na medida é a ocorrência de uma anomalia no conjunto de dados formado por séries temporais de coletas de um nó sensor ou por dados coletados pelos nós vizinhos, nas condições descritas. Nesse trabalho, ela tem carácter temporário.*

### 3.3.4 Modelo de Evento

O evento de interesse  $E$  é definido como segue.

**Definição 3.14.** *Evento: Definido no capítulo 2.1, ele é modelado como uma tupla com os seguintes elementos:*

- *raio de influência  $r$ : informa o raio (área circular) em que  $E$  pode ser detectado, sendo  $E_o$  o centro do evento.*
- *raio de influência de valor fixo  $r_f$ : informa se  $r$  varia seu valor com o tempo.*
- *velocidade da origem do evento  $v$ : informa a velocidade com a qual  $E_o$  se move, caso o evento seja móvel e ou nulo, caso contrário.*
- *aleatoriedade da ocorrência do evento  $a_o$ : descreve se a localização da ocorrência de  $E_o$  é aleatória ou determinística.*



- *aleatoriedade da trajetória do evento  $a_t$ : caso o evento seja móvel, descreve se a trajetória de  $E_o$  é aleatória ou determinística.*
- *valores de caracterização  $V_c$ : intervalo numérico de valores para os quais a característica monitorada  $j$  medida pelo sensor  $i$  representa a ocorrência de  $E$ .*
- *duração  $d$ : indica o tempo de vida de  $E$ .*
- *frequência  $f$ : fornece a frequência com que  $E$  ocorre.*

Uma aplicação para RSSFs, que tem como objetivo detectar a ocorrência de um determinado evento, pode tê-lo previamente caracterizado por um estudo, usando técnicas de aprendizado de máquina ou mineração de dados. Nesse caso, a aplicação utiliza todas as características conhecidas do evento para melhorar a qualidade de sua detecção ou mesmo prever onde ocorrerá o próximo evento.

Caso não se disponha desse estudo prévio, a aplicação pode fazer essa avaliação durante o período de tempo destinado à detecção de eventos, adaptando, assim, a aplicação a esses valores em tempo real (Rajasegarar et al., 2008). Assumimos que sabemos apenas as características  $j$  que os sensores devem monitorar e quando uma medida é considerada como uma ocorrência do evento.

Supomos que a ocorrência de  $E$  é caracterizada pela medida do valor de uma determinada característica física  $j$  dentro de uma faixa de valores conhecida e que, inicialmente, não sabemos os valores das outras informações que compõem a tupla que descreve  $E$ .

### 3.3.5 Modelo de Detecção de Eventos

Existem diversos modelos de detecção de eventos na literatura. Em Nakamura & Souza (2010), o mais simples é o modelo de detecção binária, em que, dado um nó sensor com raio de detecção  $r_d$ , ocorre a detecção caso o evento aconteça dentro da área circular descrita por  $r_d$ . Outros modelos mais sofisticados propõem uma detecção mais elaborada, baseada na distância entre o nó sensor e o centro do evento, e podem também ser vistos com maior detalhes em Nakamura & Souza (2010). Para fins de simplificação, usaremos nesse trabalho o modelo de detecção binária.

A detecção de um evento  $E$  definido pela característica física  $j$  é modelada como a detecção de um processo estocástico independente. Um processo físico, ou seja, aquele que monitora fenômenos físicos é dito ser estocástico quando pode ser mapeado para um conjunto de variáveis aleatórias indexadas por um grupo de parâmetros (Kannan, 1979). Nesse trabalho, temos o tempo discretizado  $t = 1 \dots T$ , sendo  $T$  o tempo

de vida da RSSF-OE. O processo físico é uma série temporal de variáveis aleatórias que representam as medidas coletadas da característica de interesse  $j$ . Inicialmente, consideraremos que essas variáveis aleatórias são independentes em relação ao tempo  $t$  e a posição no espaço.

Considere que o processo estudado é a variação da temperatura em um ambiente. Temos que a temperatura é uma variável aleatória dada por uma função  $T(id, t)$ , definida por:

$$\begin{aligned} T(id, t) &= X(id, t) + \alpha(X(id, t)) + \epsilon(id, t) \\ &\quad \vdots \\ T(id, t) &= X(id, t) + \alpha(id, t) + \epsilon(id, t) \end{aligned}$$

sendo que  $id$  é o identificador único de um nó sensor (que também define univocamente sua localização), e  $t$  é o momento no qual a coleta foi feita.  $X(id, t)$  é a função de distribuição de probabilidade que fornece o valor coletado da medida real da característica de interesse (*Def. 3.5*),  $\alpha(id)$  é a função que fornece o erro de medida associado a uma dada medida real (*Def. 3.6*) e  $\epsilon(id, t)$  é a função de probabilidade que fornece o erro aleatório (*Def. 3.6*). A função é associada a um determinado nó sensor em um determinado instante de tempo. O modelo de medida utilizado é uma adaptação do Modelo de *Grubbs* (*Def. 3.4*).

Em trabalhos futuros, trataremos o processo físico como um processo estocástico *Markoviano*. Esse processo possui a propriedade markoviana que garante que, em sistemas temporais, dado o estado atual, o estado passado e futuro são (condicionalmente) independentes (Kannan, 1979). Nesse trabalho, temos que o estado atual e os estados anteriores a ele são irrelevantes para prever o estado futuro.

O espaço de estados, nesse sistema de detecção, é composto pelos estados que cada nó sensor pode assumir, sendo eles: (i) “evento”, quando é detectada a ocorrência de um determinado evento, e (ii) “normal”, caso contrário. Representando a rede como uma matriz de adjacências e definindo os estados como ‘1’ quando o nó sensor detectar um evento e ‘0’, caso contrário, temos matrizes de elementos binários como possíveis configurações de estado do sistema.

A probabilidade de um nó sensor detectar um evento aumenta na proporção em que os seus nós vizinhos também o detectam. Outro fator que aumenta a probabilidade de detecção de evento por um nó sensor no estado atual  $\Delta$  é o seu estado  $\Delta - 1$ .

Os piores desempenhos para a detecção de  $E$  são esperados ocorrerem durante as transições de estado e nos limites da área de influência de  $E$ , pois utilizamos métodos de detecção que consideram a correlação espaço-temporal do conjunto de dados formado pelo nó sensor e seus vizinhos. Usaremos o modelo preditivo filtro de média móvel

(Diebold, 2007; Nakamura, 2007; Nakamura et al., 2007) para explorar a correlação temporal dos dados. Essa técnica recebe uma sequência  $n$  de dados digitais de entrada e, como sugere o nome, realiza a média desses valores. O valor  $M$  fornece o tamanho da janela deslizante, isto é, as últimas  $n$  observações realizadas são utilizadas no cálculo da média.

**Definição 3.15.** *Filtro de Média Móvel: Dada uma sequência de sinais digitais como entrada  $x = (x(1), x(2), \dots)$ , o sinal real  $y = (\hat{y}(1), \hat{y}(2), \dots)$  é estimado por:*

$$\hat{y}(k) = \frac{1}{M} \sum_{i=0}^{M-1} x(k-i) \quad (3.3)$$

sendo que  $k \geq M$ .

**Definição 3.16.** *Método de Inferência baseado em limiar: Após cada sensor ter estimado a ocorrência de um evento, essa informação é compartilhada com toda a sua vizinhança. Em seguida, cada nó sensor, em posse do resultado de detecção de seus vizinhos, decide se detectou o evento. Se mais de um determinado limiar de nós vizinhos tiverem detectado o evento, o sensor decide por considerar a ocorrência de um evento (Krishnamachari & Iyengar, 2004).*

### 3.4 Considerações Finais

Esse capítulo discutiu o escopo e o problema abordado, identificando os principais fatores envolvidos e como são modelados. Definiram-se os modelos de RSSF-OE, de medida de interesse e seus componentes, de nó sensor, de falha, de evento e sua detecção.

O desafio enfrentado por essa modelagem é o cenário limite em que não há distinção entre a ocorrência de um evento e a anomalia de dados causada por falhas. Nesses cenários, torna-se necessária uma maior amostragem de dados para inferir a ocorrência de eventos.

O próximo capítulo descreve os métodos que serão utilizados para a detecção de falhas e de eventos. Esses métodos baseiam-se na correlação espacial entre os dados coletados por nós sensores que estejam próximos fisicamente para aumentar a probabilidade de acerto no resultado da detecção. Outro assunto abordado é a extensão do arcabouço de fusão de dados *Diffuse* para o cenário de detecção de eventos (Nakamura, 2007).

# Capítulo 4

## Detecção de Falha e Eventos

Detecção de falhas e eventos é um tipo de monitoramento de ambiente que visa identificar a ocorrência de um acontecimento de interesse. Ela é baseada na probabilidade de sensores próximos fisicamente coletarem valores numéricos semelhantes.

Segundo Ding et al. (2005), a falha pode ser vista como um tipo especial de evento que contém apenas um ponto (o próprio sensor), e caracterizado pela independência espacial da ocorrência.

Diversos fatores injetam erros durante o processo de sensoriamento feito por instrumentos de medida. O ambiente possui ruídos de diferentes fontes, instrumentos de medida possuem precisão limitada e podem sofrer falhas de vários tipos durante o processo. Esses e outros fatores introduzem anomalias no conjunto de dados coletados e prejudicam o processo de detecção. Tanto para melhorar a qualidade da detecção quanto para fornecer uma metodologia sistemática de comparação de métodos, propõe-se a extensão do arcabouço *Diffuse* para esse cenário.

A seção 4.1 descreve o arcabouço de fusão de dados *Diffuse* e apresenta a sua extensão para o cenário de detecção de eventos. A seção 4.2 descreve o processo de detecção e apresenta os algoritmos gerais para a detecção de falha e evento. A seção 4.3 apresenta os métodos de detecção. A seção 4.4 faz algumas considerações finais sobre o *Diffuse* estendido e sua interferência na dinâmica de detecção de eventos.

### 4.1 Arcabouço *Diffuse*

Um arcabouço é um protótipo que serve como base para algum tipo de construção. Segundo Riehle (1969), arcabouço é um modelo de classe que define a colaboração de objetos. Ou seja, um arcabouço é uma abstração que reúne as funções em comum que colaboram para solucionar problemas que possam ser modelados de uma determinada

forma. O arcabouço de fusão de dados *Diffuse* possui um formato genérico para facilitar seu uso em diferentes aplicações, tais como detecção de eventos, rastreamento de alvo, *in-network query processing*, ou em quaisquer outras que sejam baseadas em fusão de dados (Nakamura, 2007).

Em uma avaliação do *Diffuse*, o arcabouço foi implementado para o cenário de detecção de falhas de nós sensores em uma RSSF com fluxo contínuo de dados, usando métricas de tráfego. A uma taxa constante de entrega de pacotes, o tráfego de dados poderia se modificar apenas se houvesse alteração na quantidade de sensores que constituía a rede. O roteamento de pacotes era feito utilizando uma árvore de roteamento fixa. No trabalho de Nakamura (2007), quando um nó sensor falha, ele não volta a funcionar ficando, assim, impossibilitado de reencaminhar os pacotes recebidos da sua subárvore de roteamento. Conseqüentemente, a menos que uma nova árvore seja construída, pode ser que existam nós sensores que fiquem incapacitados de se comunicarem com o nó sorvedouro. Para resolver esse problema, sempre que tal situação era identificada, a árvore de roteamento era reconstruída.

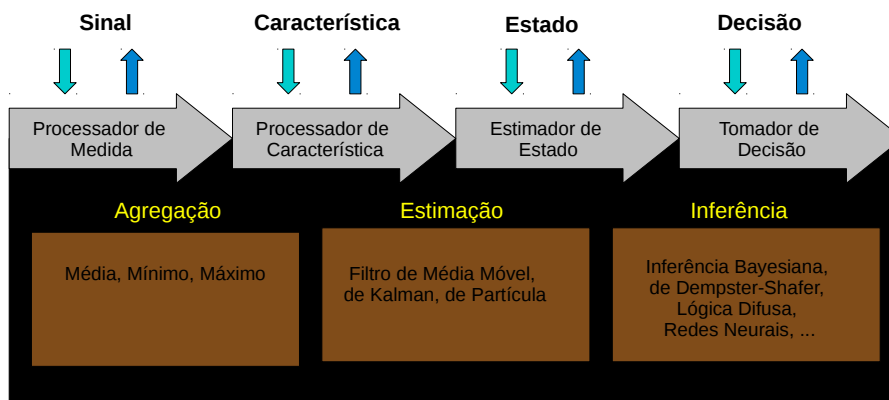


Figura 4.1: Arcabouço *Diffuse*, traduzido de Nakamura (2007).

Dado um conjunto de medidas coletadas por cada nó sensor ou compartilhadas entre os nós vizinhos, cada etapa da execução no *Diffuse* consiste em processar o dado de entrada do componente atual através de técnicas de agregação, estimação ou inferência. Dentre as opções de métodos de agregação, Nakamura (2007) cita médias, mínimos e máximos. Esses métodos podem ser empregados para sumarizar dados e para melhorar a confiança da informação enviada. Já os métodos de estimação podem ser usados tanto para filtrar dados coletados que contenham ruídos, como para estimar características de interesse. Filtro de média móvel, de Kalman e de partículas são

alguns exemplos de métodos de estimação. Outra opção de tratamento de dados são os métodos de inferência, que permitem detecções e classificações (Nakamura, 2007), sendo a inferência Bayesiana e de Dempster-Shafer opções típicas.

O *Diffuse* fornece suporte ao processo de monitoramento, que se estende desde o sensoriamento dos dados à interpretação do seu significado, indicando a melhor ação cabível a ser executada em cada situação. Ele é composto por quatro componentes, como mostra a figura 4.1: (i) Processador de Sinal, (ii) Processador de Característica, (iii) Estimador de Estado e (iv) Tomador de Decisão.

Em (i), as medidas coletadas pelo nó sensor são filtradas para melhorar a qualidade do dado coletado. Em cenários reais, toda coleta de dados, que é feita por um sensor, está sujeita a interferências do ambiente, como ruídos físicos ou eletromagnéticos, flutuações numéricas do valor da característica de interesse, imprecisão do instrumento de medição, falhas de medidas dos sensores, entre outros fatores que podem comprometer a validade dos dados coletados. A partir dos dados medidos, em (ii), transforma-se as características monitoradas em entidades que representem melhor a abordagem específica de um problema. De acordo com os valores assumidos pelas características de interesse, infere-se, em (iii), o estado atual  $\Delta$  da RSSF, dado um conjunto pré-estabelecido de estados possíveis. Finalmente, dado o estado atual  $\Delta$ , em (iv), é escolhida a ação a ser executada dado um conjunto de ações cabíveis.

O fluxo de informação inicia-se no Processador de Sinal, que recebe o dado coletado, e termina no Tomador de Decisão, que deduz que ação deve ser realizada dados o conjunto de ações possíveis e o conjunto de dados coletados.

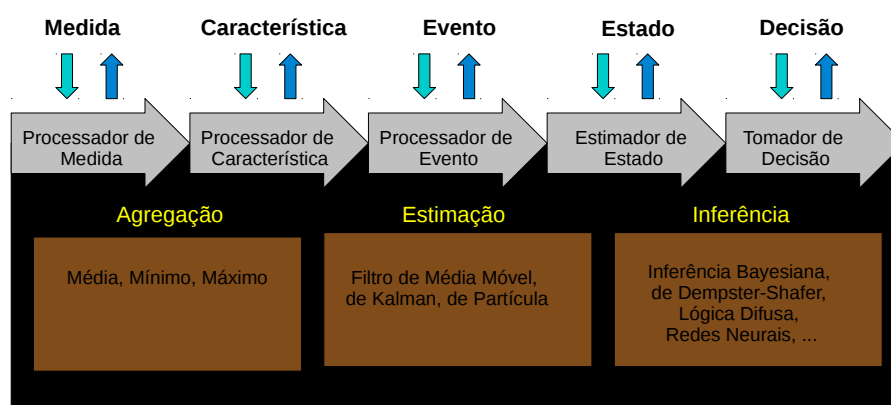


Figura 4.2: Arcabouço *Diffuse* estendido para o cenário de detecção de eventos.

Independente da escolha do filtro, do método de estimação ou de inferência, o

*Diffuse* pode ser estendido para tratar detecção de eventos, como ilustrado na figura 4.2. Nesse caso, os seus componentes passam a ter as seguintes funções:

- **Processador de Medida:** os dados são filtrados para a retirada de parte do ruído inserido pelo ambiente. A seguir, eles são sumarizados com base na correlação espacial-temporal, com a finalidade de identificar a ocorrência de falhas de nós sensores.
- **Processador de Característica:** os dados pré-processados são transformados em entidades que identifiquem melhor a ocorrência do evento de interesse.
- **Processador de Evento:** com os dados transformados, cada nó sensor decide localmente se houve a ocorrência de um evento.
- **Estimador de Estado:** os estados possíveis na RSSF-OE são “evento” ou “normal”. Esse módulo estima o estado  $\Delta$  de cada sensor a partir de um conjunto de detecções vizinhas ou ao longo de um intervalo de tempo, empregando métodos de inferência.
- **Tomador de Decisão:** de posse do estado atual  $\Delta$ , esse componente decide encaminhar uma notificação ao nó sorvedouro em caso de detecção positiva.

O diagrama da figura 4.3 detalha a adaptação do *Diffuse* para o cenário de detecção de eventos. Inicia-se com a coleta de dados pelos nós sensores. Esses dados passam pelo Processador de Medida que é composto por três etapas: filtragem de dados, envio dos dados filtrados para os nós vizinhos e detecção de falhas de nós sensores. Na última etapa, caso o sensor detecte que falhou, ele substitui sua própria medida pela sumarização das coletas feitas pela vizinhança. Caso contrário, ele continua normalmente com a etapa de Processamento de Característica. Caso necessário, o sensor transforma o dado coletado em uma entidade que facilite o processo de detecção, caso contrário segue para a etapa de Processamento de Evento. Essa fase é constituída de duas partes: estimar a ocorrência de evento, e enviar o resultado da detecção para os vizinhos. A seguir, de posse da detecção de todos os nós vizinhos, cada sensor infere seu estado. Caso o estado seja “evento”, envia uma notificação para o nó sorvedouro; caso contrário, reinicia a coleta de dados.

## 4.2 Processo de Detecção

O processo de detecção de eventos possui três etapas: (i) coleta, (ii) processamento de estado e (iii) decisão. Os nós sensores coletam informações sobre as caracte-

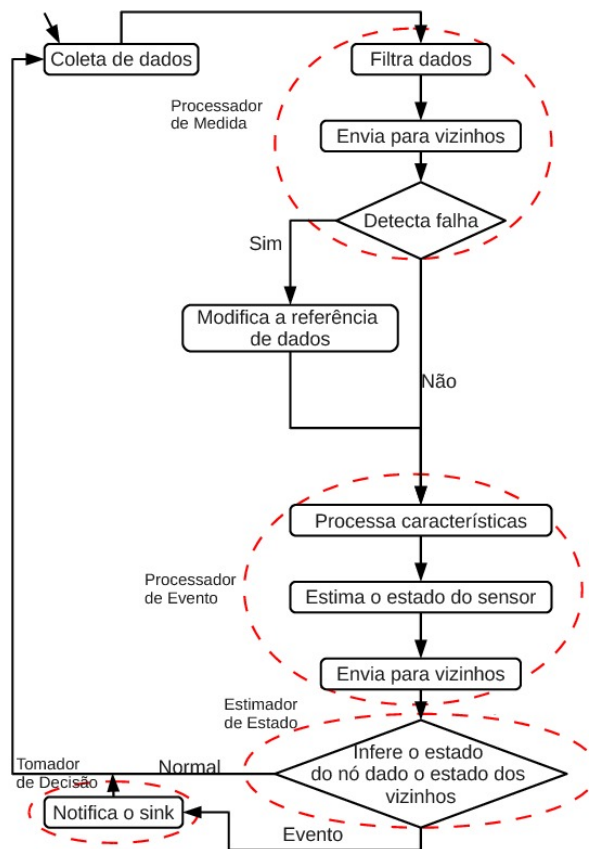


Figura 4.3: Esquema de implementação do *Diffuse* estendido para o cenário de detecção de eventos.

terísticas de interesse do ambiente, processam esses dados para, então, inferirem sobre a ocorrência de eventos. Seguindo os moldes fornecidos pelo *Diffuse* estendido (seção 4.1), a etapa (i) envolve decisões de modelagem do cenário de interesse, a etapa (ii) engloba desde o processamento de medida até a estimação do estado da rede e a etapa (iii) é representada pela tomada de decisão final, podendo incluir a ação de nós atuadores.

Com relação ao cenário de sensoriamento, esse trabalho explora eventos simples identificados por uma única característica de interesse, cuja medida é feita por um instrumento sensor não-ideal sujeito a falhas durante o processo. A característica possui um valor que flutua ao longo do tempo de acordo com uma distribuição *Gaussiana* com média igual ao valor real. A modelagem completa do problema é descrita em detalhes no capítulo 3.

Um sistema de detecção de eventos precisa responder às seguintes perguntas:



- Como distinguir dados coletados entre medições reais e falhas?
- Como inferir a ocorrência de eventos a partir dos dados considerados reais?

Empregando técnicas que sejam capazes de responder essas perguntas, é possível desenvolver diversos tipos de aplicações.

Essa seção aborda o processo de detecção de falhas e eventos de modo geral. Primeiramente, apresenta-se como são feitas as comparações de intervalos que serão usadas nos métodos propostos. Em seguida, expõe-se como é o processo em comum de detecção de falhas e eventos abordado nesse trabalho.

### 4.2.1 Comparação de Intervalos

Os métodos propostos são baseados em comparação de intervalos e correlação espacial. A medida feita por um dispositivo sensor segue o Modelo de *Grubbs*, fornecendo um intervalo numérico ao invés de um valor pontual (vide seção 3.3.1). Já o conjunto formado pelos valores coletados por nós vizinhos é sumarizado em um intervalo numérico pelos métodos que serão apresentados na Seção 4.3.

Como esse trabalho considera a ocorrência de falhas de medida, é necessário pré-processar os dados coletados, separando-os em grupos de dados corretos e incorretos, para melhorar a eficiência do sistema de detecção.

Tanto durante a detecção de falhas quanto na de eventos, há separação dos dados em dois grupos usando a comparação de intervalos numéricos. O intervalo  $i_n$  que define a medida feita por dado sensor é comparado com dois outros intervalos: um intervalo  $i_m$ , que caracteriza o quão representativo é o dado coletado em relação aos seus vizinhos (usado para detectar falhas); e outro intervalo  $i_E$  que identifica se o dado coletado é um evento.

No processo detecção de falhas, existem três casos a considerar:

1.  $i_n \subset i_m$ : existe coerência entre o valor medido pelo nó sensor e o valor sumarizado dos dados coletados pelos seus vizinhos. Nesse caso,  $i_n$  é considerada como uma medida correta.
2.  $i_n \not\subset i_m$ : o dado do sensor não se correlaciona fortemente com os dados dos vizinhos, fornecendo um indicativo da ocorrência de uma falha durante o processo de medição.
3. No caso intermediário, em que  $i_n$  não está completamente contido em  $i_m$ , a medida é tratada como dado correto pelos métodos descritos nesse capítulo.

Sejam  $[m_1, m_2]$  e  $[n_1, n_2]$  os intervalos que representam  $i_m$  e  $i_n$ , respectivamente. Sem perda de generalidade, suponha também que  $\rho(i_m) > \rho(i_n)$ , sendo  $\rho(X)$  uma função que devolve o tamanho do intervalo definido por  $X$ . A seguir, consideraremos os três casos possíveis de relacionamento entre esses intervalos: a certeza da validade dos dados (caso 1), a certeza da ocorrência de falhas (caso 2), e dados parcialmente confiáveis (caso 3).

No caso 1, há uma grande chance de os dados coletados serem corretos, pois  $i_n \subset i_m$ . A caracterização desse caso é  $m_1 < n_1 < n_2 < m_2$ , como podemos ver na figura 4.4.

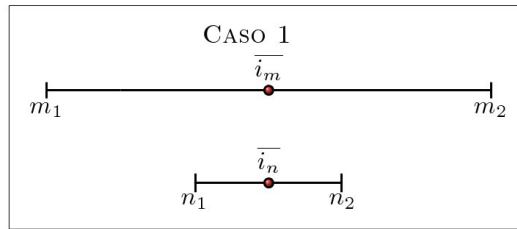


Figura 4.4: Não há falhas no sensoriamento, já que  $i_n \subset i_m$ , i.e., o valor coletado pelo nó sensor é coerente com os valores coletados pelos seus nós vizinhos.

Já no caso 2, existem dois subcasos, como vemos na figura 4.5. Em ambos, percebe-se que  $i_n \not\subset i_m$ . Adicionalmente, no subcaso 2.1, tem-se:

$$n_1 > m_2 \quad (4.1)$$

E no subcaso 2.2 tem-se:

$$n_2 < m_1 \quad (4.2)$$

Nesse caso, ambas as possibilidades indicam uma grande chance de o dado coletado se tratar de uma falha.

Finalmente, o caso 3 abrange a situação em que não se garante a ocorrência ou não de uma falha, como podemos ver na figura 4.6. Esse caso é formado por dois subcasos. O subcaso 3.1 caracteriza-se por:

$$m_1 < n_1 < m_2 < n_2 \quad (4.3)$$

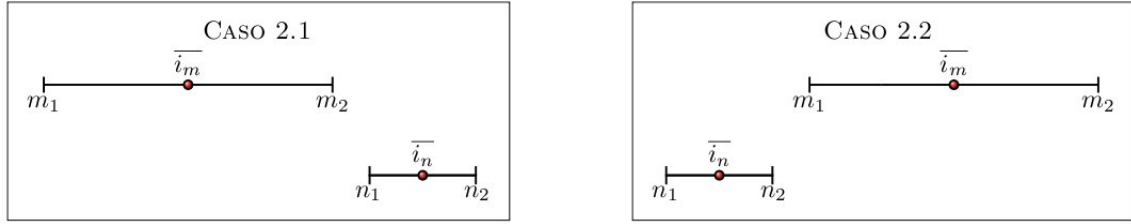


Figura 4.5: Há falhas no sensoriamento já que  $i_n \not\subset i_m$ , i.e., o valor coletado pelo nó sensor não é coerente com os valores coletados pelos seus nós vizinhos.

e o subcaso 3.2 é determinado por:

$$n_1 < m_1 < n_2 < m_2 \quad (4.4)$$

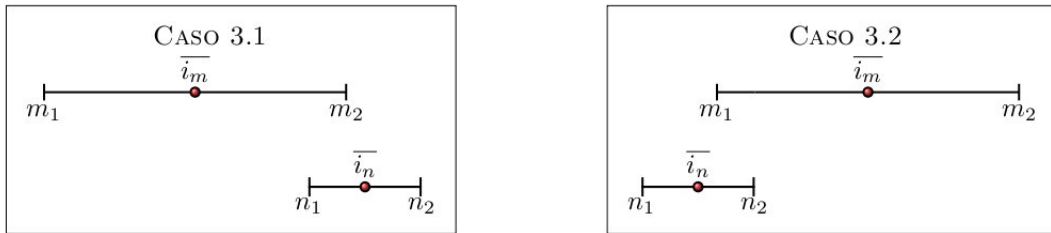


Figura 4.6: Não sabemos se houve falha durante o sensoriamento.

Nos dois últimos subcasos, há dúvidas sobre como se relacionam os intervalos  $i_n$  e  $i_m$  e, portanto, não se sabe a princípio se houve falha. Para decidir quando ocorre um erro nesses casos, pode-se diminuir o intervalo  $i_m$  até que  $i_n$  esteja completamente dentro ou fora dele. Observa-se que no processo de comparação de dois intervalos, a confiança final na detecção da ocorrência de falhas é, no máximo, igual a menor confiança dos intervalos. Antecipa-se que  $i_n$  estará dentro de  $i_m$  caso a média que define  $i_n$  também estiver dentro. Nessa situação, haverá falha se tivermos:

$$\rho(i_m \cap i_n) \geq k\rho(i_n), \quad (4.5)$$

sendo  $0 \leq k \leq 1$ . Assim,  $k$  é o fator experimental que define quando  $i_n$  é aceito como

medida real, a partir do percentual de  $i_n$  que foi interceptado por  $i_m$ .

Suponha que o evento  $A$  seja “a medida real está dentro do intervalo  $i_n$ ” e o evento  $B$  seja “a medida real está dentro do intervalo  $i_m$ ”. É possível calcular a confiança máxima e mínima que se pode ter nesse processo de decisão de ocorrência de falhas, usando as fórmulas:

$$P(A \cap B) = P(A) + P(B) - P(A \cup B) \quad (4.6)$$

e

$$P(A \cap B) = P(B | A)P(A) \quad (4.7)$$

Supondo que  $\rho(i_n) < \rho(i_m)$ , pode-se analisar os dois casos extremos de intersecção entre dois intervalos: (i)  $i_n \subset i_m$  e (ii)  $i_n \not\subset i_m$ . No primeiro caso, se o valor real pertence a  $i_n$ , ele também pertencerá a  $i_m$ . Usando a equação 4.7 tem-se:

$$P(B | A) = 1 \quad (4.8)$$

e, portanto,

$$P(A \cap B) = P(A) \quad (4.9)$$

Já no segundo caso,  $P(A \cup B)$  pode ser no máximo igual a 1, e usando a equação 4.6 tem-se:

$$P(A \cap B) = P(A) + P(B) - 1 \quad (4.10)$$

Assim, a probabilidade do valor estar em  $i_n$  e  $i_m$  está entre estes dois limites.  $P(A \cap B)$ , que é também a confiança que se tem em dizer que não ocorreu uma falha, é portanto resumida por:

$$P(A) + P(B) - 1 \leq P(A \cap B) \leq P(A) \quad (4.11)$$

Quanto maior a confiança ao medir os dados, menor o intervalo de valores que  $P(A \cap B)$  pode assumir, e melhor a qualidade da triagem dos dados. Deduz-se que ocorreu falha caso a média que especifica  $i_n$  não pertença a  $i_m$ . É, portanto, tem-se uma medida real no caso contrário.

Dessa forma, o conjunto de dados é considerado confiável e será utilizado para inferir a ocorrência de um evento. A comparação de intervalos, agora, é feita entre  $i_n$  e  $i_E$ , o intervalo que caracteriza a ocorrência de um evento. Novamente, sem perda de generalidade, assume-se que  $\rho(i_E) > \rho(i_n)$ . Para determinar se o valor coletado pelo nó sensor é um evento, identificam-se os mesmos grupos de casos verificados anteriormente

para  $i_n$  e  $i_m$ :

- $i_n \subset i_E$ : ocorrência de um evento.
- $i_n \not\subset i_E$ : não ocorrência de um evento.
- $i_n$  intercepta  $i_E$ , mas não está contido nele: nesse caso, não se tem certeza se ocorreu ou não um evento.

No último caso, a mesma análise apresentada para  $i_n$  e  $i_m$  é processada, agora relacionando  $i_n$  e  $i_E$ , para decidir se um evento ocorreu. Se durante a detecção de falha, a medida do sensor for classificada como falha, comparamos  $i_m$  e  $i_E$  durante a detecção de evento.

## 4.2.2 Detecção de Falha

Nesse trabalho, falhas são tratadas como um tipo especial de evento (Palpanas et al., 2003). Assim, os métodos propostos são usados tanto para detectar falhas quanto eventos.

Com base na discussão sobre a comparação de intervalos (seção 4.2.1), e na correlação espacial das medidas de nós sensores vizinhos, propomos o algoritmo 11 para determinar quais nós da RSSF possivelmente mensuraram um valor falho.

```

1: for all  $n \in V$  do
2:   Coleta  $i_n$ , a medida da característica física  $j$ .
3:   Envia a medida por broadcast para os seus vizinhos.
4:   Sumariza os valores recebidos dos vizinhos em  $i_m$ .
5:   Identifica a relação entre  $i_n$  e  $i_m$ .
6:   if  $i_n \not\subset i_m$  then
7:      $i_n =$  Falha de Medida
8:   else
9:      $i_n =$  Medida Real
10:  end if
11: end for

```

O algoritmo 11 é executado em cada nó sensor da rede. Ele usa o valor da característica física de interesse  $j$  coletada pelo sensor. Essa medida possui erros  $\epsilon_{ij}$  associados ao ruído e associados à imprecisão  $\alpha_i$  do instrumento. Assim, cada sensor constrói o intervalo  $i_n = x_j + \epsilon_{ij} \pm \alpha_i$ , como visto na definição 3.4. Cada instrumento envia  $i_n$  para os seus vizinhos e calcula o intervalo de sumarização  $i_m = [m_1, m_2]$  do

método a partir dos valores coletados pelos vizinhos. Nesse trabalho, esse intervalo pode ser dado pelo gráfico de controle de qualidade ou pelo intervalo de confiança. Então, o sensor identifica a relação entre  $i_n$  e  $i_m$ , inferindo se houve uma falha.

### 4.2.3 Detecção do Evento

Se se desejar explorar as particularidades de um cenário, é necessária uma estratégia diferente para a detecção de cada tipo de evento. No entanto, esse trabalho trata eventos de uma maneira geral.

O algoritmo 9 apresenta a detecção de eventos, sendo executado em cada nó sensor da rede, tal como o algoritmo 11. Após determinar se sua medida é uma falha, cada nó sensor identifica se sua medida (ou o intervalo de sumarização do método, caso sua medida seja uma falha) representa a ocorrência de um evento.

```

1: for all  $n \in V$  do
2:   Detecta se houve falha de medida
3:   Identifica a relação entre  $i_n$  (ou  $i_m$ , se houve falha) e  $i_E$ , intervalo que caracteriza
   a ocorrência de evento.
4:   if  $i_n \notin i_E$  then
5:      $i_n = \text{normal}$ 
6:   else
7:      $i_n = \text{evento}$ 
8:   end if
9: end for

```

Esse trabalho trata a detecção de um evento simples e bem comportado, conforme descrito a seguir. Em intervalos de tempo fixos, cada nó sensor coleta informações do ambiente em que está inserido para inferir a ocorrência do evento  $E$ . Cada conjunto de dados coletados em um determinado instante relaciona-se com o conjunto de dados coletados anterior e posteriormente (correlação temporal). Além disso, os nós sensores próximos fisicamente coletam dados relacionados (correlação espacial). Assim, espera-se que quanto menor o intervalo de tempo entre coletas de dados, menor a diferença entre os valores dos dados coletados, sendo que a mesma observação pode ser feita para a distância entre os nós sensores vizinhos.

No cenário abordado nesse trabalho, há uma forte correlação espacial e temporal dos dados coletados pelos sensores e seus vizinhos, como acontece em muitos cenários de RSSFs (Nakamura, 2007; Nakamura et al., 2007).

O sistema de detecção coleta dados do ambiente a uma taxa de amostragem  $S$  que garante a existência da correlação temporal entre os dados. Cada nó sensor tem a capacidade de armazenar uma sequência  $n$  de amostras que são compartilhadas com os nós vizinhos.

A duração do evento ultrapassa o intervalo de amostragem. Assim garantimos que temos, pelo menos, uma amostra do evento durante a sua ocorrência. O cenário ideal seria aquele em que o evento tem um tempo de vida muito maior que  $S$ , de forma que teríamos uma base que representaria melhor a ocorrência de eventos e poderíamos explorar essa característica em métodos de detecção.

Nesse trabalho, a identificação do evento é binária, ou seja, cada sensor detecta ou não um evento dentro do seu raio de sensoriamento. Todo evento  $E$  possui um raio de influência  $r$ , centrado em  $E_o$ , e a medida do sensor possui o mesmo valor em qualquer ponto dentro da área circular dada por  $r$ . A modelagem do problema é descrita em detalhes no capítulo 3.

A priori, sabe-se qual é o evento que se quer monitorar e como identificá-lo. A caracterização da ocorrência do evento é feita quando uma medida coletada está dentro de um dado intervalo numérico. Li et al. (2008) afirmam que a lógica para o uso de limites de valores como ocorrência de eventos é que quando eventos ocorrem, acontecem mudanças detectáveis em algumas características físicas do ambiente monitorado. Então, um evento pode ser capturado uma vez que o valor de uma característica sensoriada atinja um determinado limiar.

Outro aspecto importante é a determinação dos limites da detecção do evento, i.e., até onde o evento é detectável pela RSSF. Se não houvesse falhas de nós sensores, uma simples consulta a cada nó sensor da vizinhança resolveria o problema. Mas em um cenário com falhas de medidas, os dados coletados pelos nós sensores não são confiáveis. Assim, os métodos de detecção usam técnicas baseadas em correlação espacial e temporal dos dados, supondo que exista baixa ocorrência de falhas.

Nesse trabalho, cada nó sensor possui somente um único dispositivo sensor que coleta uma única característica do ambiente. Sem perda de generalidade, será adotado um modelo de amostras independente e identicamente distribuído (*iid*) para as observações coletadas por cada nó sensor ao longo do tempo.

A comunicação entre sensores é feita via rádio. Os raios de sensoriamento e de transmissão de dados dos sensores são iguais e não diminuem com o passar do tempo. Cada sensor sabe quem são seus vizinhos usando um protocolo de *broadcast* com limitação de um salto de comunicação.

A modelagem do cenário sugere um ambiente controlado, tal como uma estufa ou um laboratório, em que se deseja controlar a temperatura, por exemplo. Assim,

apenas variações suaves nas medições da temperatura são permitidas, sendo dadas pela função de probabilidade que descreve o comportamento da característica de interesse, pela limitação de precisão dos instrumentos sensores e pelo ruído do ambiente. Nesse sentido, esse trabalho relaciona métodos que focam na atenuação dos efeitos do ruído e das falhas de nós sensores nas decisões de detecção. Assume-se ainda que as falhas e o ruído são estocasticamente independentes (Luo et al., 2006).

Projetou-se tal cenário como o mais simples possível, uma vez que a intenção desse trabalho é verificar a eficácia dos métodos propostos, utilizando uma extensão do arcabouço *Diffuse*. Apesar de simples, esse cenário possui grande potencial para aplicações de monitoramento. Pode-se utilizar esse esquema de detecção mesmo em aplicações que não permitem esse tipo de deposição de nós sensores, desde que se garanta uma quantidade mínima de nós vizinhos.

### 4.3 Métodos Propostos

Essa seção propõe dois métodos de detecção de falhas e de eventos para a implementação do *Diffuse* estendido. Foram selecionados os seguintes métodos:

- **Processador de Medida:** Foi utilizado um filtro de média móvel para melhorar a qualidade dos dados coletados em relação ao ruído. Para detectar falhas de sensores, foi usado um dos métodos descritos.
- **Processador de Característica:** Por tratarmos de eventos univariados, a característica coletada é a melhor aborgagem para o tratamento do problema e não precisamos transformá-la algebricamente.
- **Processador de Evento:** Para detectar um evento, foi implementado um dos métodos propostos.
- **Estimador de Estado:** Para estimar o estado de um nó sensor, foi utilizado um método baseado em limiar. Essa etapa refina a detecção de eventos da rede. O conjunto de estados possíveis é ‘ocorrência’ e ‘não-ocorrência’ de evento.
- **Tomador de Decisão:** O conjunto de ações possíveis é: notificar o nó sorvedouro em caso de detecção positiva e continuar o processo de monitoramento, em caso contrário.

O filtro de média móvel é fácil de entender e implementar, sendo esse um dos motivos de sua popularidade entre os métodos de detecção de eventos. Por isso, nesse



trabalho, ele será implementado como um filtro de medida no Processador de Medida. Esse tipo de filtro é recomendado quando há ruído randômico branco durante a coleta de dados (Def. 3.15).

Como a probabilidade de erros de detecção varia com o cenário de simulação, será utilizado um método baseado em limiar para a inferência de ocorrência de eventos no Estimador de Estado.

Essa seção descreve os métodos propostos. Eles são usados em duas etapas: (i) detecção de falhas e (ii) detecção de eventos como dito na seção 4.2. A seguir, apresentamos o Método do Intervalo de Confiança que é uma abordagem estatística para a detecção de eventos em RSSF. E, em seguida, apresentamos um método baseado em Gráfico de Controle de Qualidade.

### 4.3.1 Método do Intervalo de Confiança

O Método do Intervalo de Confiança (Confidence Interval Method – CIM) baseia-se na correlação espacial das medidas coletadas. Cada nó sensor verifica o quanto sua última medida aproxima-se das coletas feitas por sua redondeza. Cada nó sensor calcula o intervalo de confiança usando os valores coletados pelos seus vizinhos, obtendo-se  $i_m$ . O  $i_m$  serve como um intervalo referencial para detecção de falha. O intervalo de confiança é dado por

$$\mu \pm t_{[1-\frac{1}{\alpha};n-1]} \left( \frac{\sigma}{\sqrt{n}} \right) \quad (4.12)$$

sendo  $\mu$  e  $\sigma$ , a média e o desvio padrão do conjunto de dados formado pelas coletas vizinhas;  $n$  o tamanho da amostra e  $\alpha$  o nível de significância. O valor de  $t_{[1-\frac{1}{\alpha};n-1]}$  é tabelado para amostras de tamanho até 30.

O  $\sigma$  fornece uma medida da qualidade dessa amostra. Quanto menor é o valor de  $\sigma$ , melhor é a qualidade da amostra, i.e., mais correlacionada espacialmente ela é e, portanto, mais certeza pode-se ter da ocorrência ou não de um evento  $E$ .

A figura 4.7 apresenta um fluxograma que detalha a implementação do Método do Intervalo de Confiança (CIM). A primeira etapa é a coleta de um *streaming* de dados e o uso do filtro de média móvel para a melhoria da qualidade explorando a correlação temporal. Em seguida, esses dados são compartilhados com os vizinhos que estão a um salto de distância e é aplicado o método de detecção de falhas (CIM). Ele consiste em calcular o intervalo de confiança (IC) do conjunto de dados formado pelos vizinhos do sensor (não inclui o dado coletado pelo próprio sensor que calcula o IC). Comparando  $i_n$  e  $i_m$ , o método decide se  $i_n$  é uma falha. Caso seja, ele é substituído pelo valor de  $i_m$ .

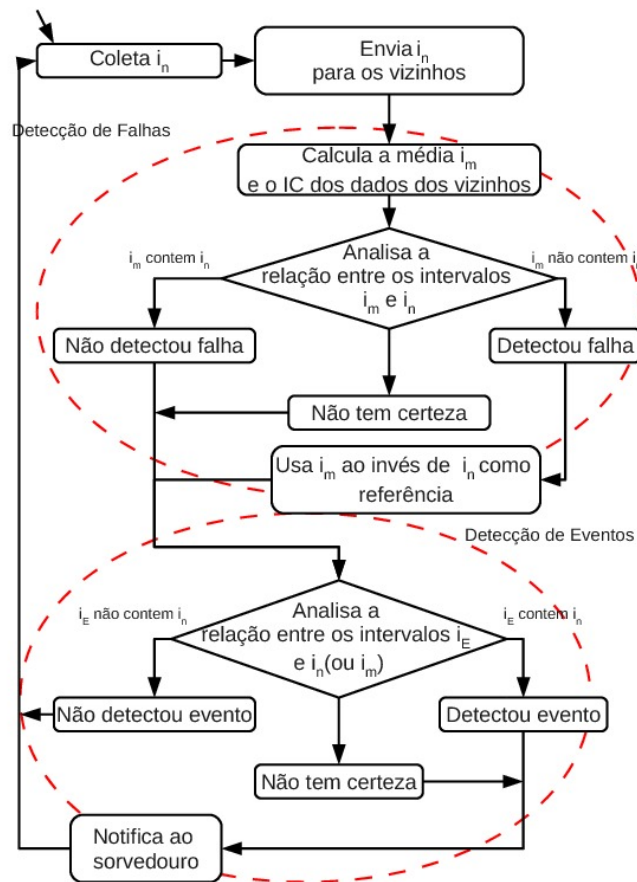


Figura 4.7: Fluxograma do Método do Intervalo de Confiança tolerante a falhas.

Na sequência, é aplicado o método de detecção de eventos com o uso de comparação de intervalos ( $i_n$  – ou  $i_m$  caso tenha sido detectada uma falha – e  $i_E$ , intervalo que caracteriza a ocorrência de evento), determinando o estado da rede: “ocorrência” ou “não-ocorrência” de um evento. A detecção do evento é, então, compartilhada com a vizinhança. Cada nó sensor decide, com base no conjunto de detecções fornecidas pelos vizinhos, qual é sua decisão final sobre a ocorrência local do evento.

Nas simulações-padrão, se mais de 50% dos vizinhos concordarem que houve uma ocorrência de evento, o nó sensor assume que ele aconteceu e notifica ao nó sorvedouro, senão retoma o processo. Esse fluxograma é executado continuamente durante todo o tempo de vida da RSSF.

### 4.3.2 Método de Gráfico de Controle de Qualidade

O Método de Gráfico de Controle de Qualidade (Control Chart Method – CCM) é um método de detecção baseado na Teoria de Gráficos de Controle de Qualidade (*Control Charts*). Segundo (1989), Costa et al. (2005) e Ekambaram (1972), esse tipo de controle estatístico consegue inferir quais das medidas coletadas pelos sensores têm maior probabilidade de ser uma falha ou um evento.

Segundo Oliveira (2008), o Gráfico de Controle de Qualidade é uma técnica de monitoramento *online* para identificar eventos que modificam um processo de fabricação industrial. O gráfico descreve algum cálculo estatístico feito sobre um conjunto de amostras ordenadas ao longo do tempo.

Existem diferentes gráficos de controle na teoria de Controle Estatístico do Processo dada por (1989) e Ekambaram (1972). Os gráficos de controle dividem-se em gráficos que monitoram (i) dados de atributo e (ii) os que monitoram dados variáveis. No primeiro caso, é necessário contar quantas vezes um determinado evento aconteceu. Por exemplo, em processos industriais, deseja-se saber quantos tubos trincados apareceram na linha de produção durante um determinado período de tempo. Para os problemas em que se quer contar uma determinada característica do ambiente ou de um objeto, usam-se os Gráficos de Controle de Atributos. Mas, em outros casos, precisa-se saber qual o tamanho médio das trincas e não a quantidade. Para tal, monitoram-se as dimensões de uma certa característica do ambiente ou do produto que está sendo fabricado, usando o segundo tipo de gráfico – os Gráficos de Controle de Variáveis.

Os processos podem ser classificados de acordo com suas características específicas: (i) se a quantidade de amostras, tomadas segundo alguma frequência de tempo, é constante; e (ii) se é necessário contar eventos ou medir dimensões. Para cada caso, há um conjunto específico de gráficos de controle para monitorar o processo.

O primeiro passo para controlar estatisticamente um processo é identificar o que o constitui, qual é a característica de interesse e qual o tipo de gráfico de controle que se deve escolher para realizar o monitoramento. Nesse trabalho, deseja-se identificar quando ocorre um evento de interesse. A ocorrência de um evento acontece quando um nó sensor, que coleta dados de uma determinada característica do meio, captura um dado dentro de um determinado intervalo de valores. Como se pretende coletar medidas de uma característica para determinar a ocorrência de um evento é natural escolher um dos Gráficos de Controle de Variáveis.

O gráfico de controle que será utilizado é o  $\bar{X}$ . Utiliza-se esse gráfico para fornecer tanto a informação de quais dados da amostra são, provavelmente, inválidos e constituem uma falha, quanto para identificar a ocorrência de um evento. Em cada

sensor, coleta-se um *streaming* de medidas a cada instante de tempo ao invés de uma única medida. Calcula-se a média dessas medidas que é então inserida na Carta  $\bar{X}$ . Essa etapa do processo é equivalente ao Processador de Medida do arcabouço *Diffuse* estendido.

Esse gráfico possui três linhas de controle que auxiliam na identificação de dados fora do controle do processo. A Linha Central é uma linha horizontal de altura igual à média das médias das medidas. A linha de Limite Superior de Controle (LSC) e a Linha Inferior de Controle (LIC) situam-se, respectivamente, acima e abaixo da Linha Central. A Tabela 4.1 informa como são calculadas essas linhas para a Carta  $\bar{X}$ .

<i>Gráfico de Controle</i>	<i>Linha Central</i>	<i>LSC</i>	<i>LIC</i>
$\bar{X}$	$\mu_{\bar{X}}$	$\mu_{\bar{X}} + 3\sigma_{\bar{X}}$	$\mu_{\bar{X}} - 3\sigma_{\bar{X}}$

Tabela 4.1:  $\sigma_X$  é o desvio padrão do conjunto de dados e  $n$  é o tamanho da amostra.

A maioria dos processos industriais segue uma distribuição Gaussiana. Segundo (1989), mesmo que a distribuição do processo não seja descrito por uma Gaussiana, a distribuição das médias do processo tendem a ela quando a quantidade de medidas aumenta.

Ainda segundo (1989), uma distribuição Gaussiana possui cerca de 68%, 95% e 99,7% dos valores contidos no intervalo limitado por  $k\sigma_X$ , sendo que  $k = \{1, 2, 3\}$ , respectivamente. Assim sendo, a probabilidade de se ter algum valor fora do intervalo formado por  $\mu_X \pm k\sigma_X$  é de 0,3%, tal como ilustra a figura 4.8.

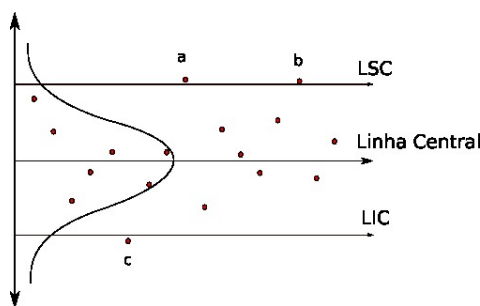


Figura 4.8: Probabilidade das medidas estarem fora das linhas de controle e serem dados corretos é de 0,3%. Os pontos a, b e c tem 99,7% de probabilidade de serem dados inválidos e serão descartados.

Nesse trabalho, o processo de detecção de evento segue a mesma descrição de medidas fora de controle que os processos industriais, ou seja, baixa frequência de

acontecimento. Supomos que a característica monitorada tenha mudanças suaves de valores, i.e., temos um evento bem comportado, pois “embora a oportunidade de ocorrência de um defeito seja infinita, a probabilidade de ocorrência em um certo lugar ou instante é muito pequena” (Costa et al., 2005).

O método de detecção de eventos usando gráficos de controle, assim como o Método do Intervalo de Confiança, é constituído por duas etapas: (i) identificação de falhas de medidas e (ii) detecção de ocorrência de eventos, como se pode perceber na figura 4.9. Na primeira etapa, cada nó sensor, de forma síncrona, coleta um conjunto de dados, que denominaremos de amostra, e calcula a média  $\mu$ . Em intervalos de tempo igualmente espaçados, coletam-se amostras. As médias das amostras coletadas pelo nó sensor são encaminhadas para os nós vizinhos. Cada sensor recebe as médias dos seus vizinhos e as distribui em sua Carta  $\bar{X}$ . Em seguida, o sensor calcula as linhas de controle.

Caso o conjunto de dados coletados pelos vizinhos não estiver dentro dos limites impostos por [LCI, LCS], eliminam-se os dados que estiverem fora dos limites e recalculam-se as linhas até que todo o conjunto de dados usados para calculá-las estejam dentro dos limites [LCI, LCS]. Quando esse estado for atingido, comparam-se os intervalos  $i_n$  e [LCI, LCS] para detectar se a medida feita pelo sensor é válida. Caso não seja, o valor de  $i_n$  é substituído por [LCI, LCS] durante a etapa de detecção de eventos.

Esse gráfico de controle indica quais dos dados coletados, possivelmente, são falhas, pois segundo Ekambaram (1972) e (1989), dados que estiverem fora dos limites impostos pelas Linhas de Controle Superior e Inferior têm 99,7% de probabilidade de estarem errados, podendo, então, serem descartados. Toda vez que um dado é descartado pode haver mudanças nos valores das Linhas de Controle. É necessário, portanto, recalculá-las e reavaliar o conjunto de pontos até que todos estejam contidos dentro dessas linhas.

O segundo passo é identificar, a partir desse conjunto de dados pré-selecionados, se eles caracterizam a ocorrência de um evento de interesse. Visualizando o problema sob essa ótica, pode-se usar o gráfico de controle  $\bar{X}$  da mesma forma em que se usa no processo industrial. A Carta  $\bar{X}$  visa verificar se o conjunto de medidas está dentro dos valores considerados sob controle. Quando se percebe uma nova tendência do conjunto de dados, redefine-se o conjunto de linhas de controle.

Durante o processo de detecção de eventos, a rede oscila entre dois estados: “ocorrência” e “não-ocorrência” de evento. A mudança de estado durante esse estágio – um salto de um estado para o outro – caracteriza a mudança na tendência do conjunto de amostras. Caso não esteja acontecendo um evento e a mudança da tendência dos dados

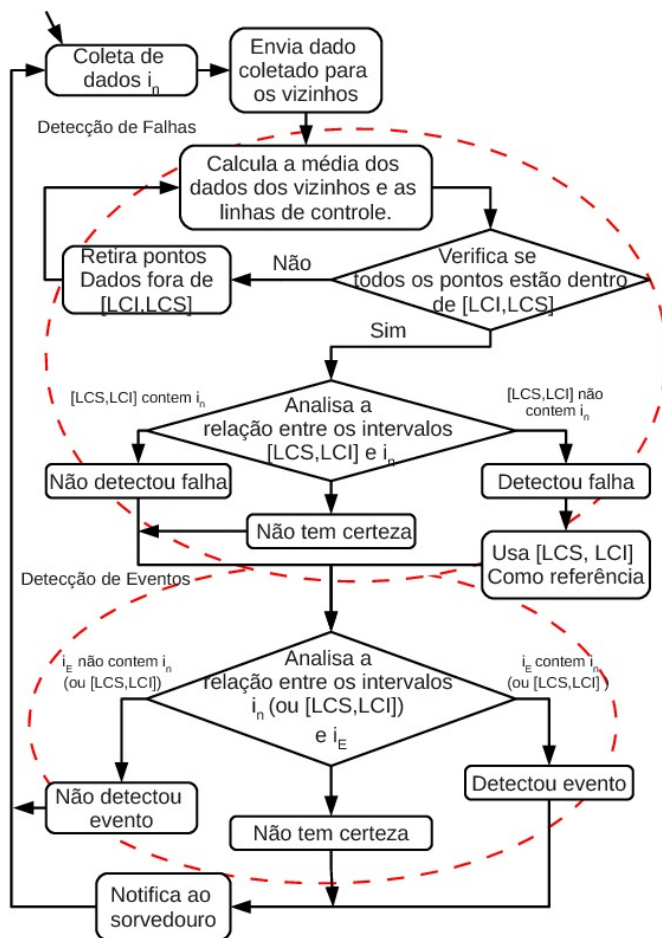


Figura 4.9: Fluxograma do algoritmo de detecção usando o Método de Gráfico de Controle de Qualidade.

fornecer uma média dentro de  $i_E$ , aconteceu um evento. Caso esteja acontecendo um evento e a mudança na tendência dos dados fornecer uma média fora de  $i_E$ , o evento terminou. Como mensagens são enviadas ao nó sorvedouro apenas quando se identifica a ocorrência de um evento, a mudança no estado da detecção implica na inversão do estado de envio – se a rede está enviando dados para o nó sorvedouro, ela pára de enviá-los, e caso contrário, começa a enviá-los.

Segundo Costa et al. (2005), quando uma sequência de cinco pontos consecutivos ou mais está acima ou abaixo da linha central, tem-se uma indicação de que o processo não está controlado, tal como ilustra a figura 4.10. Nesse caso, pode ter acontecido uma mudança no comportamento do processo, que, agora, se comporta com média e desvio padrão diferentes.

Nesse trabalho, o que caracteriza a ocorrência de um evento são os valores coleta-

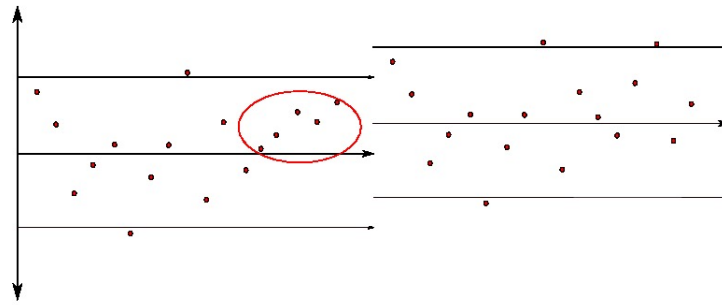


Figura 4.10: Quando cinco pontos consecutivos ou mais estão acima ou abaixo da linha de controle central, o processo está fora de controle. Quando isso acontece, é necessário recalculer as linhas de controle a partir desse conjunto de dados.

dos pelo nó sensor e pelos seus vizinhos. O momento de detecção de eventos é quando as Linhas de Controle Inferior e Superior definem um intervalo que está contido no intervalo que caracteriza a ocorrência de eventos.

## 4.4 Considerações Finais

Esse capítulo contextualiza o problema central da dissertação. Detecção de eventos é um tema que pode ter diversas abordagens, o que dificulta a comparação entre os trabalhos e, portanto, a eleição do estado da arte do tema, como visto no capítulo 1. Assim sendo, o arcabouço *Diffuse* estendido em conjunto com os modelos-padrão dados no capítulo 3 fornece uma metodologia sistemática para comparação de combinações de técnicas de detecção de eventos para aplicações reais, já que admitimos que detecção é um processo estocástico.

# Capítulo 5

## Simulação e Discussão dos Resultados

Este capítulo apresenta os resultados de simulação para os métodos propostos no capítulo anterior. A seção 5.1 descreve a ferramenta de simulação usada na experimentação. A seção 5.2 descreve os experimentos realizados para avaliar os métodos propostos bem como o arcabouço adaptado *Diffuse*. Também descreve o cenário padrão, os parâmetros, os fatores e as métricas envolvidas. A seção 5.3 apresenta os resultados de simulação e análise para os métodos Gráfico de Controle de Qualidade e Intervalo de Confiança. Finalmente, a seção 5.4 discute os métodos propostos e suas diferentes configurações.

### 5.1 Ferramenta de Simulação

O *Sinalgo* é um arcabouço de simulação usado para testar e validar algoritmos de rede (Sinalgo, 2011). Ou seja, o *Sinalgo* abstrai as camadas da rede e foca em simular o algoritmo.

Ele é baseado em um esquema de *steps* e *rounds*. Os nós sensores são criados no início da simulação e mantêm as suas características durante todo o tempo de vida da rede que é dividido em *rounds*. Cada *round* é composto por um conjunto de *steps* que serializa as suas atividades. Cada *round* possui um período inicial e final de propósito geral, atualização de conexões entre os nós e entrega de mensagens.

Apesar de ser um simulador de algoritmos, o *Sinalgo* permite a implementação de problemas particulares em redes. É possível a implementação tanto de tipos específicos de nós sensores e mensagens usadas para comunicação entre os dispositivos, quanto dos modelos usados. Modelos de conectividade, de mobilidade e de distribuição de nós na



área de deposição, de interferência, de confiabilidade e de transmissão de mensagens também podem ser implementados. Além da capacidade de comunicação, cada nó da rede pode processar dados coletados ou recebidos.

O simulador pode executar em dois modos: síncrono e assíncrono. No primeiro caso, é simulada uma sequência de passos considerando que todos os nós da rede processam os dados ao mesmo tempo. Já no segundo modo, o processamento é baseado na ocorrência de eventos. Este trabalho usa o modo síncrono.

O *Sinalgo* possui um conjunto de implementações padrão e projetos pré-existentes em Java. Ao criar novos projetos nesse simulador, podemos reutilizar estas implementações ou qualquer outra biblioteca da API Java.

## 5.2 Descrição do Experimento

A rede é depositada sob a forma de uma grade regular na área de interesse. A área de deposição não possui obstáculos físicos que poderiam atrapalhar o funcionamento dos canais de comunicação. Com isso, este tipo de deposição provê a garantia da cobertura total da área de interesse e a de uma quantidade fixa de vizinhos.

Assume-se que todo nó sensor possui um raio de comunicação fixo. Isto permite que ele se comunique apenas com seus oito ou 24 vizinhos mais próximos, dependendo da simulação. Este trabalho não trata nenhum tipo de perda de sinal nem de erros associados a transmissão de pacotes.

As simulações foram executadas no simulador de algoritmos *Sinalgo*. O simulador trata a rede como um grafo em que cada nó é um sensor que possui as capacidades de processamento de dados e de comunicação com os outros sensores com os quais possui arestas.

A tabela 5.1 apresenta os modelos fornecidos pelo simulador que foram utilizados nas simulações. A escolha dos modelos já implementados são feitas no arquivo de configuração `Config.xml`. Além do uso destes modelos, foram criados dois modelos de nós sensores, um para cada método proposto.

O objetivo das simulações é verificar a eficácia de cada método proposto, avaliando as métricas relativas à qualidade da detecção. A eficácia dos métodos de detecção de eventos é avaliada de acordo com a taxa de detecção positiva (verdadeiro positivo) e negativa (verdadeiro negativo) de eventos.

Entende-se por taxa de detecção positiva de evento quando na ocorrência de um evento, ele é detectado. Já a taxa de detecção negativa é quando na não-ocorrência de um evento, ele não é detectado. As taxas de detecção são um valor entre 0 e 1 e

<i>Modelo</i>	<i>Valor</i>
Modelo de Mobilidade	<i>NoMobility</i>
Modelo de Transmissão de Mensagens	<i>ConstantTime</i>
Modelo de Interferência	<i>NoInterference</i>
Modelo de Conectividade	<i>StaticUDG</i>
Modelo de Deposição	<i>grid2D</i>
Modelo de Confiabilidade	<i>ReliableDelivery</i>

Tabela 5.1: Modelos usados nas simulações.

representam a porcentagem de sensores que realizaram a detecção corretamente.

O complemento da taxa de detecção positiva é o erro falso negativo ou erro do tipo I, assim como, o complemento da taxa de detecção negativa é o erro falso positivo ou erro do tipo II. Os algoritmos propostos foram avaliados em relação às taxas positivas e negativas de detecção de falhas, de eventos e de inferência de eventos.

Foram identificados os seguintes parâmetros de simulação dos métodos propostos:

- quantidade de vizinhos de um nó sensor,
- número de sensores na rede,
- tamanho da área de interesse de deposição,
- tipo de ruído,
- valor da medida do sensor caracterizada pela ocorrência de evento e pela não-ocorrência de evento,
- desvio padrão da medida do sensor,
- valor de uma medida do sensor com falha,
- tipo de filtro de medidas e tamanho da janela temporal de medidas usada no filtro,
- tipo de medida realizada pelo sensor,
- precisão do aparelho de medida do sensor,
- raio de comunicação do sensor,
- porcentagem mínima de sensores vizinhos que detectaram um evento para que ele seja inferido,

- porcentagem de nós sensores da rede que falham,
- distribuição dos sensores falhos na rede,
- tipo de falha,
- modelos utilizados do *Sinalgo*,
- semente de números pseudo-aleatórios usada na amostragem de distribuições usadas nos métodos (distribuição de falhas e amostragem da medida e de ruído)
- intervalo de confiança usado nos métodos propostos (inerente aos métodos),
- tempo de vida da rede,
- tempo de duração do evento,
- raio e posição do evento e
- quantidade de eventos.

Desse conjunto de parâmetros, foram escolhidos os fatores de simulação nos níveis dados pela tabela 5.2. Os outros parâmetros foram assinalados com os valores dados pela tabela 5.3.

	<i>Fator</i>	<i>Valor</i>
1	Quantidade de Vizinhos	8 e 24
2	Desvio Padrão da Medida	{0; 2; ...; 10; 12}
3	Porcentagem de Nós que Falham	0%, 10%, 20% e 30%
4	Intervalo Confiança usado nos Métodos Propostos	95% e 99,5% ou 99,7%
5	Métodos de Detecção	Método Gráfico de Controle de Qualidade e Método Intervalo de Confiança
6	Porcentagem de Sensores Vizinhos usados Durante a Inferência de Eventos	{ 0,3; 0,4; ...; 0,7}

Tabela 5.2: Fatores usados nas simulações.

Os eventos foram gerados utilizando o ambiente e linguagem R. Tanto o raio quanto a posição  $x$  e  $y$  do centro do evento circular foram dados pela função *runif()* do R. O raio de comunicação dos sensores varia para ajustar a quantidade de vizinhos que se deseja avaliar.

	<i>Parâmetros</i>	<i>Valor</i>
1	Número de Sensores	1024
2	Área de Deposição	$100 \times 100 \text{ m}^2$
3	Tipo de Ruído	Ruído Randômico Branco
4	Valor da Medida do Sensor na Área de Ocorrência de Evento	33,0
5	Valor da Medida do Sensor na Área de Não-Ocorrência de Evento	10,0
6	Valor da Medida do Sensor com Falha	[100, 150]
7	Tamanho da Janela Temporal de Medidas	5
8	Medida Realizada pelo Sensor	Distribuição Gaussiana
9	Precisão de Medida do Sensor	0,05
10	Raio de Comunicação do Sensor	5 ou 9 (depende da quant. de viz.)
11	Distribuição das Falhas dos Sensores na Rede	Distribuição Uniforme
12	Tipo de Falha	Temporária (duração de 100 <i>rounds</i> )
13	Semente de Número Pseudo-Aleatório	77654767234 ( <i>Sinalgo</i> )
14	Tempo de Vida da Rede	5000 <i>rounds</i>
15	Tempo de Duração do Evento	100 <i>rounds</i>
16	Raio do Evento	[25,50](Distribuição Uniforme)
17	Posição do Evento	[0,100](Distribuição Uniforme)
18	Quantidade de Eventos	50

Tabela 5.3: Parâmetros usados nas simulações.

Os modelos que não são previstos pelo simulador e compõem o problema abordado são descritos no capítulo 3. Dentre eles, temos os modelos de medidas, de falhas de sensores, do ruído, de evento e de detecção.

O *Sinalgo* não possui um modelo de sensoriamento de dados. Por causa disso, foi implementado o Modelo de *Grubbs* (Grubbs, 1973) que modela a medida de uma característica de interesse como a soma de sua medida real, do erro do instrumento de medida e do ruído de fundo.

Em cada simulação uma porcentagem fixa de nós sensores falham. Em intervalos de 100 *rounds*, a configuração de distribuição de falhas de nós sensores na rede é modificada. Em cada configuração da distribuição de falhas, diferentes nós irão falhar e os nós que tinham falhado na configuração anterior voltarão a fazer medidas corretamente. Assim, tem-se um cenário de falhas temporárias.

O ruído randômico branco é um dos componentes formadores da medida do sensor (seção 3.3.1). O ruído é branco quando possui independência espacial-temporal. Isto é, o valor do ruído em um dado sensor é independente do valor do ruído em outro sensor e de suas ocorrências anteriores.

O arcabouço adaptado *Diffuse* foi configurado como se segue:

- Processador de Medida: Filtro de Média Móvel com janela temporal igual a cinco.
- Processador de Falha: Método de Detecção de Falha igual ao Método de Detecção de Evento.
- Detector de Evento: Método Intervalo de Confiança ou Método Gráfico de Controle de Qualidade.
- Estimador de Estado: Método de Inferência baseado em Limiar.

### 5.3 Resultados e Análise

O processo de detecção de eventos executa, na sequência, detecção de falhas, detecção de eventos e inferência de eventos. Na detecção de falhas, o nó dito falho substitui sua medida pela média das medidas dos seus vizinhos. Caso a detecção de falhas tenha sido correta, o novo valor representará melhor o cenário real, melhorando a detecção de eventos. Caso contrário, a média pode representar pior o cenário real já que há uma maior probabilidade de ocorrência de falhas no conjunto de medidas dadas pelos nós vizinhos do que na medida feita por um único sensor.

Durante as simulações foram identificadas algumas propriedades dos métodos provenientes da representação das medidas:

1. Quanto maior os intervalos envolvidos na detecção maior a probabilidade de acerto das taxas positivas. Essa detecção é baseada na intersecção de intervalos numéricos. Assim, quanto maior os intervalos maior a probabilidade deles se intersectarem. O contrário acontece com as taxas negativas: maiores intervalos tendem a piorar este tipo de taxa.
2. A porcentagem de falhas impacta mais que o valor do desvio padrão da medida. A existência de medidas com falhas no conjunto de dados impacta mais no tamanho do intervalo de referência que o desvio padrão da medida. Isto acontece porque os intervalos gerados têm sua média mais deslocada em direção ao valor da falha e possuem um desvio padrão maior quando possuem falhas no seu conjunto de dados do que quando o desvio padrão das medidas aumenta.
3. A qualidade da detecção de eventos impacta diretamente na qualidade de inferência de eventos. Isso porque, a inferência usa o resultado da detecção para ser processada.

4. Com o aumento da porcentagem de falhas, aumenta a quantidade de dados falhos no conjunto de dados. Um único nó que tenha feito uma medida falha interfere em até oito conjuntos de dados, pois, na simulação padrão, há a troca de dados entre os oito vizinhos mais próximos.

Esta seção apresenta o conjunto de experimentos descritos na seção 5.2. O Método Intervalo de Confiança foi simulado para as variações com 95% e 99,5% de intervalo de confiança e o Método Gráfico de Controle de Qualidade para 99,7% de intervalo de confiança. Para esses métodos foram avaliados os impactos do desvio padrão da medida realizada pelos sensores na qualidade da detecção de falha e de evento assim como na sua inferência. Foi avaliado também o impacto da quantidade de falhas e do número de vizinhos usados nos métodos. O Método de Inferência baseado em Limiar foi avaliado variando-se a quantidade mínima de vizinhos que confirmem a ocorrência de um evento em um nó sensor.

### 5.3.1 Método Gráfico de Controle de Qualidade

Este método baseia-se na Teoria de Gráficos de Controle de Qualidade a qual é usada para o controle de processos industriais. O gráfico usado foi a Carta  $\bar{X}$  que é constituída por três linhas de controle: a linha central, a linha de controle superior e a linha de controle inferior.

Segundo essa teoria, as médias das medidas distribuídas no gráfico tendem à distribuição gaussiana com o aumento da quantidade de médias usadas. Assim sendo, com 99,7% de certeza, podemos dizer que uma média que está fora dos limites impostos pelas linhas de controle inferior e superior é uma falha.

#### 5.3.1.1 Desvio Padrão da Medida e Porcentagem de Falhas

A figura 5.1 mostra o comportamento das taxas de detecção e de inferência para variações do desvio padrão  $\sigma$  da medida dos nós sensores e da porcentagem de nós falhos na rede. Com o aumento de  $\sigma$  para uma determinada porcentagem de nós falhos, percebe-se que as taxas negativas de detecção e de inferência de eventos caem. Isto se deve à primeira propriedade mencionada: a taxa negativa é prejudicada pelo aumento dos intervalos envolvidos quando se aumenta o  $\sigma$ .

O aumento de  $\sigma$  não impacta as taxas de detecção de falhas. Ambos os intervalos envolvidos nesta detecção, a medida do sensor e a média das medidas dos vizinhos, possuem tamanho variável e podem deslocar-se livremente, o impacto do aumento é atenuado.

Ainda em relação à variação de  $\sigma$ , pode-se perceber um suave declínio da taxa de detecção positiva de eventos. O conjunto de dados formado pelas coletas feitas pelos vizinhos é refinado até que as restrições sejam cumpridas. Este efeito é o impacto do conjunto final de dados nos nós sensores que estão na borda do evento.

Inicialmente, usa-se o conjunto formado pelas coletas de todos os vizinhos para a construção do gráfico de controle. Os dados recebidos tendem para uma distribuição gaussiana com média na linha central, ou seja, aproximadamente metade dos dados deverá estar desta linha.

Caso isso não ocorra, este conjunto de dados provavelmente possui dados falhos que devem ser eliminados antes da construção do intervalo de referência definitivo. A identificação destes possíveis dados falhos é processada até que os dados selecionados para formarem o conjunto se distribuam da maneira descrita e dentro do intervalo limitado pelas linhas de controle.

Outra melhoria no conjunto de dados usados para construir o intervalo de referência é encontrar o subconjunto de tamanho  $k - 1$  que tenha o menor desvio padrão. Este conjunto final é usado para construir o gráfico de controle definitivo e, conseqüentemente, o intervalo de referência para a validade dos dados. Com o conjunto de dados definido, identifica-se a ocorrência de falhas. Este conjunto de dados impacta fortemente na taxa positiva de detecção de falhas porque, durante a triagem de dados, quando mais da metade dos dados são falhas, o método identifica as falhas como dados reais e os dados reais como falhas. Portanto, a detecção de falhas é fortemente dependente do modo como os nós falhos se distribuem na área de interesse.

Após sucessivas retiradas de dados anômalos, os nós que estão na borda do evento podem ter um intervalo de referência formado por um conjunto de dados sem valores referentes à ocorrência de eventos. Isto ocorre porque este tipo de nó possui um conjunto inicial de dados heterogêneos e sujeito a uma maior triagem. Portanto, nesta situação, o nó infere que sua medida é uma falha e assume como sua nova medida, o valor da média dos vizinhos. Esta média sugere que não houve evento naquela área ocorrendo uma falha na detecção positiva de eventos.

Analisando o aumento da porcentagem de nós falhos na rede, verifica-se que a taxa de detecção de falhas positivas decresce. Já a taxa de detecção de falhas negativas não sofre o mesmo efeito, mantendo-se praticamente constante. O aumento do número de nós sensores que fazem uma medida falha aumenta também a probabilidade dos nós da rede terem uma falha no seu conjunto de dados. Quando a quantidade de falhas ultrapassa a metade do conjunto, mesmo quando ocorre uma falha, ela nunca será detectada. Isto aumenta a taxa de detecção negativa e diminui a positiva.

O uso do método de inferência de evento após o método de detecção melhora

a qualidade dos resultados para este cenário. Cada nó sensor verifica quais dos seus vizinhos detectaram eventos. Este método diminui a taxa de erros do tipo I e II, diminuindo também o efeito de borda, onde a detecção é pior por causa da variabilidade dos dados fornecidos pelos vizinhos.

### Método Gráfico de Controle de Qualidade com 99,7% de Confiança

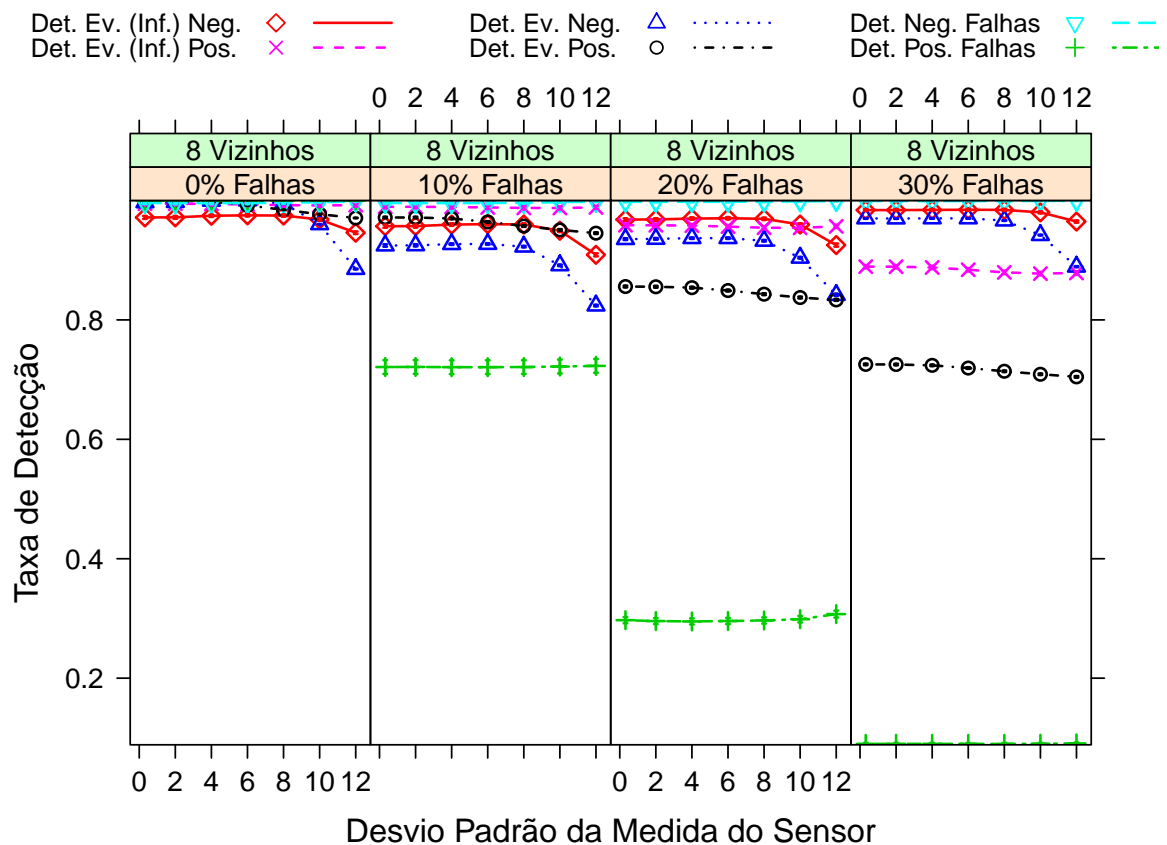


Figura 5.1: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade, variando-se a porcentagem de falhas. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.



### 5.3.1.2 Quantidade de Vizinhos, Desvio Padrão da Medida e Porcentagem de Falhas

A figura 5.2 apresenta o resultado das simulações para variações do  $\sigma$  da medida, da quantidade de vizinhos com os quais se compartilha dados coletados e da porcentagem de nós na rede que falham. Nos cenários com 0% e 10% de nós falhos, os resultados das simulações são similares para os casos em que cada nó sensor recebe dados dos seus oito vizinhos (os que estão a um salto de distância) e dos seus 24 vizinhos (os que estão a dois saltos de distância). A figura 5.3 mostra, em mais detalhes, os desempenhos da detecção e da inferência de eventos. Para os parâmetros usados, os dois cenários possuem um desempenho similar.

A figura 5.4 mostra os resultados das simulações que avaliam as taxas de detecção e inferência usando oito e 24 vizinhos em uma rede com 20% e 30% de nós falhos. Para as simulações usando oito vizinhos, a taxa positiva de detecção de falhas é melhor para os dois cenários simulados.

Na figura 5.5, pode-se observar em detalhes as outras taxas de detecções. Percebe-se uma pequena melhora nos resultados para as detecções negativas nas configurações para 24 vizinhos. Em contrapartida, as detecções positivas são ligeiramente melhores para as simulações que usam apenas oito vizinhos.

Mantendo-se a porcentagem de nós falhos fixa, quanto maior o tamanho da vizinhança, maior a probabilidade de ocorrência de falhas no conjunto de dados recebidos. Assim, para os cenário que utilizam dados fornecidos por 24 vizinhos, o efeito das falhas é intensificado em relação ao respectivo cenário com vizinhança de tamanho oito.

### 5.3.1.3 Porcentagem de Vizinhos Usada no Método de Inferência de Eventos

A figura 5.6 mostra como a inferência de eventos se comporta quando se varia a porcentagem de vizinhos usada para deduzi-la e a porcentagem de nós sensores que falham na rede. Na figura, as detecções positiva e negativa de eventos aparecem apenas como referências. O cenário padrão de simulação deste método usou 50% dos vizinhos como limiar de inferência de eventos. Isto é, o nó sensor infere que ocorreu um evento na região em que ele está, caso mais de 50% dos vizinhos o detectem.

A figura mostra o comportamento das taxas positiva e negativa de inferência de eventos. Quanto maior a porcentagem de vizinhos usados na inferência melhor é a taxa negativa. Isto ocorre porque quanto maior a exigência de concordância com a detecção dos vizinhos, menor é a quantidade de erros do tipo I (falso positivo). No entanto, mais nós sensores inferem a não-ocorrência de eventos, mesmo estando em área de ocorrência positiva, aumentando a taxa de ocorrência de erros do tipo II (falso

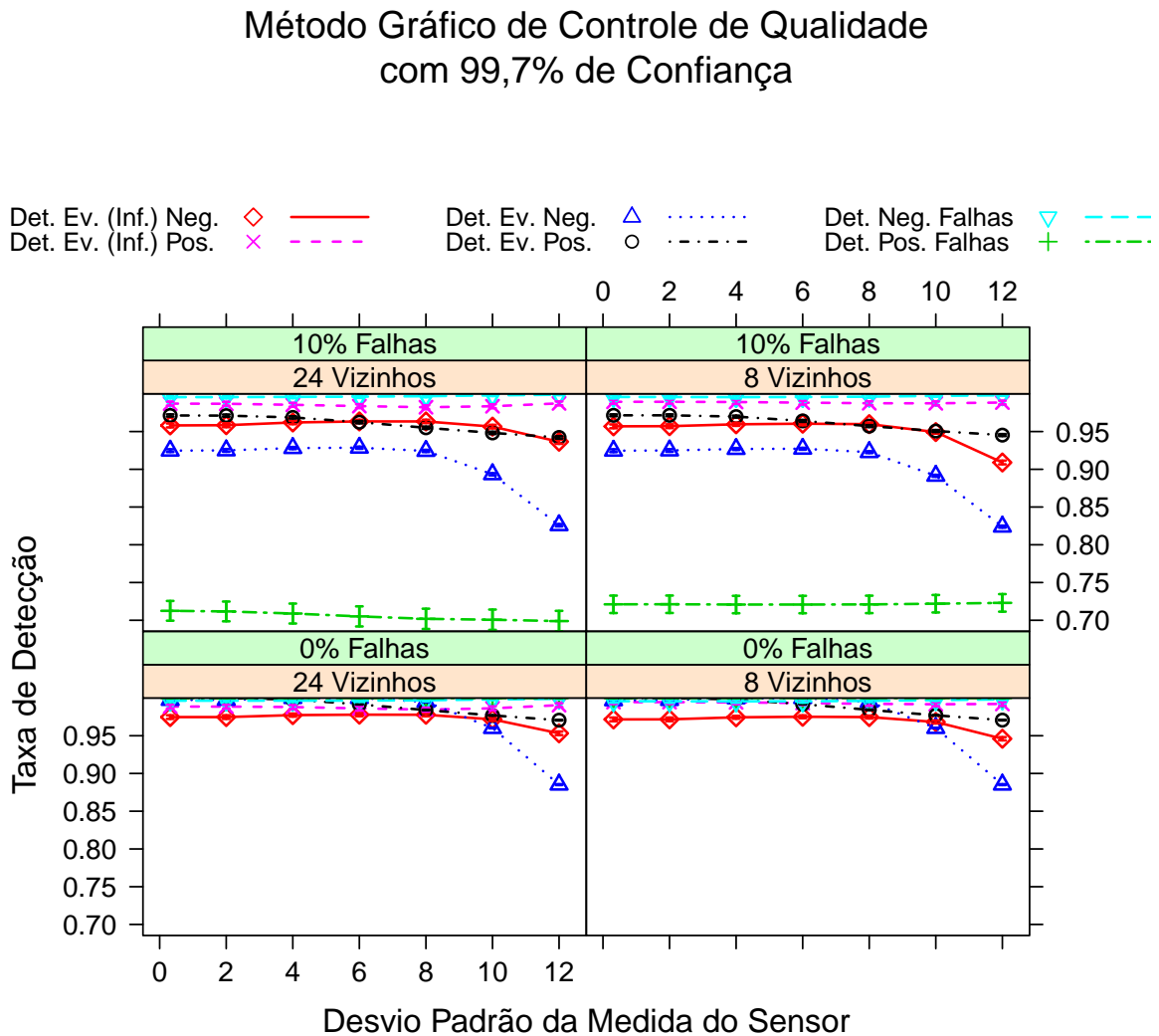


Figura 5.2: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, com 0% e 10% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

negativo) e, portanto, diminuindo a taxa positiva de inferência.

O uso de métodos de inferência de eventos é válido apenas quando o resultado obtido é melhor que o da detecção de eventos. No cenário em que não há ocorrência de falhas de sensores, o método de inferência não melhora visivelmente os resultados. No entanto, quanto maior a porcentagem de nós falhos, maior o ganho que se obtém

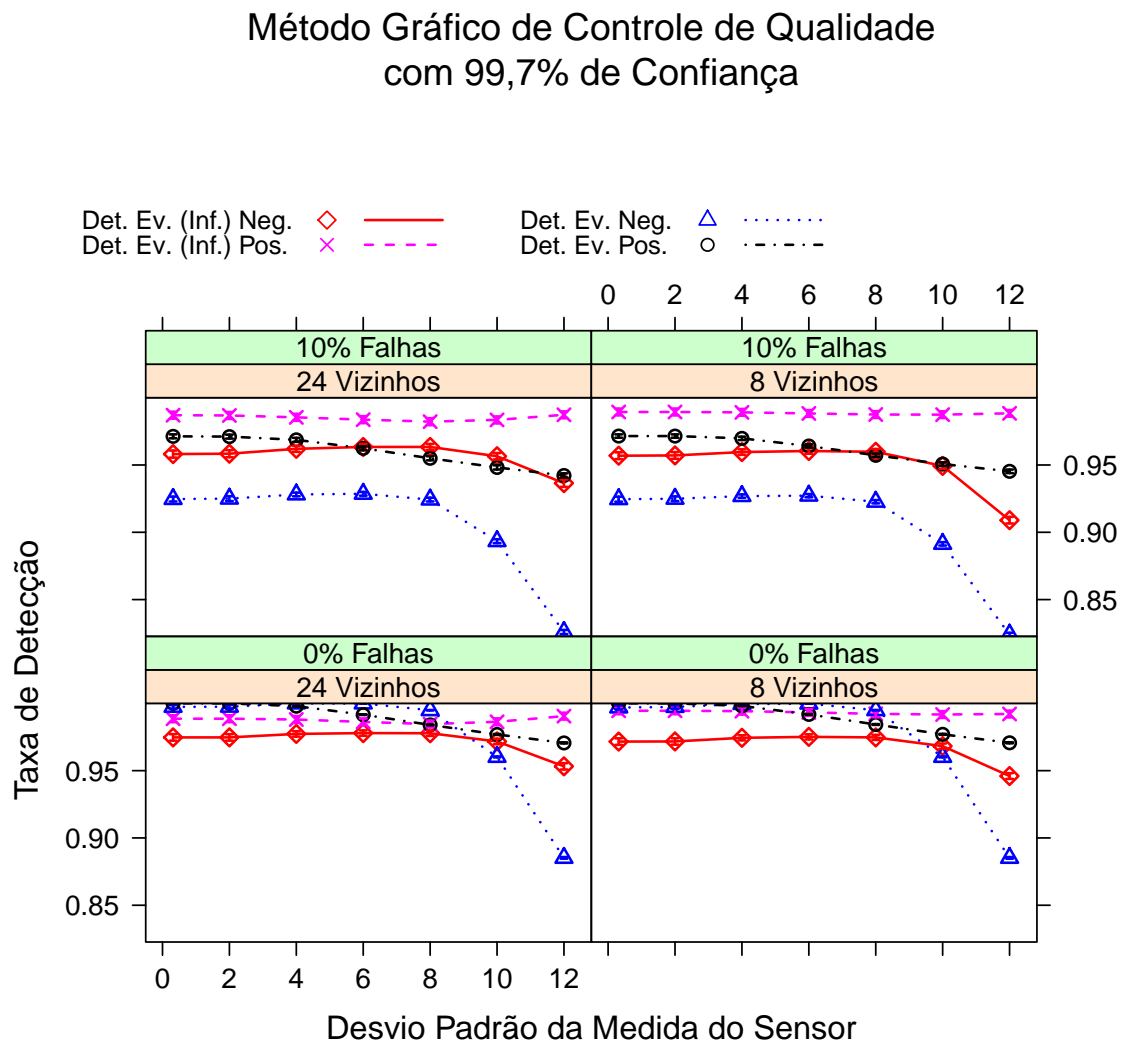


Figura 5.3: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, com 0% e 10% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

com o uso do método de inferência.

A figura 5.7 mostra o impacto nas taxas de detecção e inferência devido ao aumento da porcentagem de nós falhos na rede. As taxas positivas caem com o aumento da porcentagem de nós falhos e o efeito oposto pode ser percebido nas taxas negativas. O conjunto de dados com falhas possuem um efeito duplo no intervalo de referência:

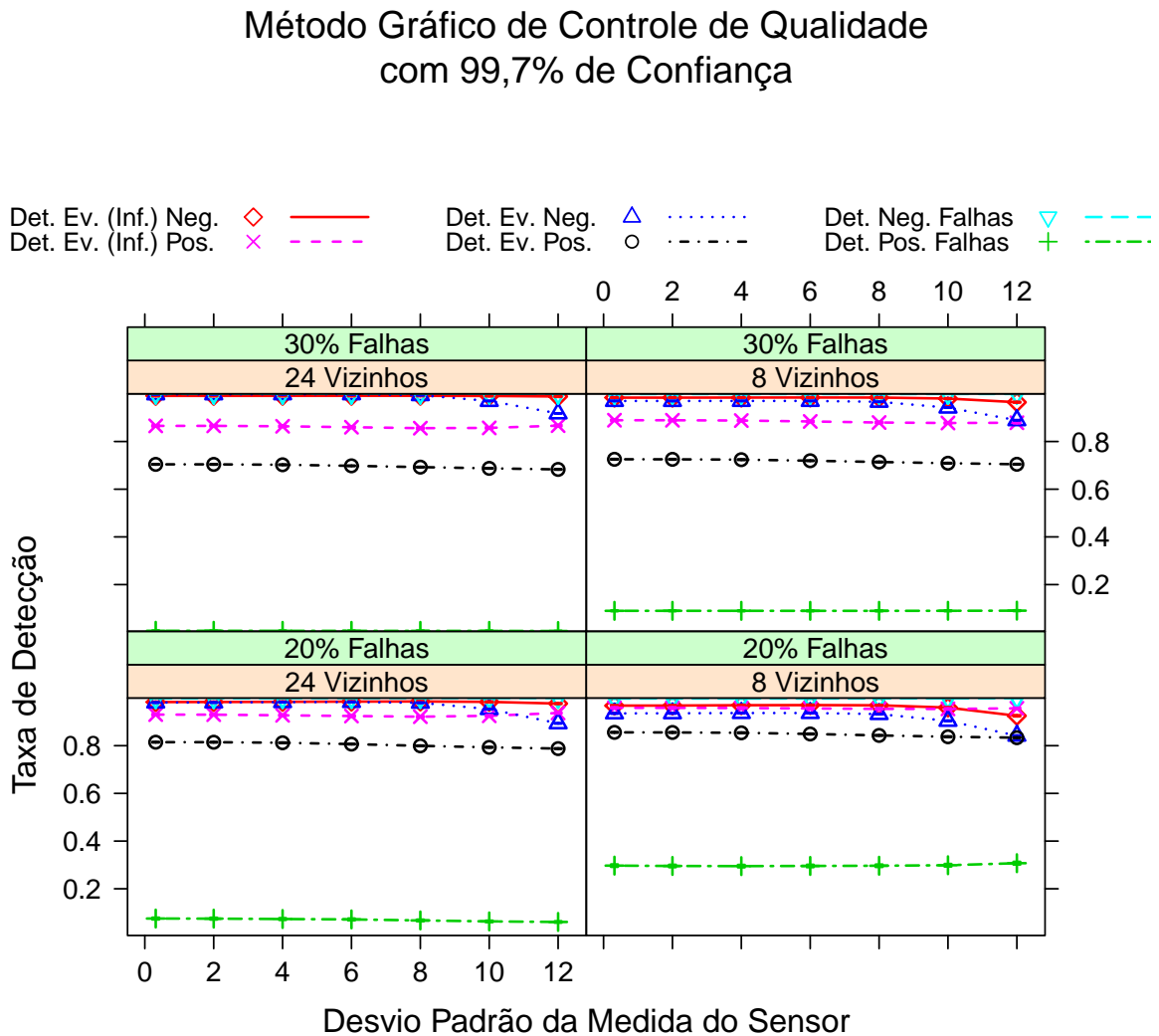


Figura 5.4: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, com 20% e 30% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

desloca a média do intervalo em direção do valor da falha e aumenta o desvio padrão, aumentando o tamanho dos intervalos. No entanto, este método proporciona um conjunto de dados final com nenhuma falha – caso o conjunto de dados tenha mais da metade dos dados válidos – ou composto apenas por falhas – caso contrário. Com o aumento da quantidade dos conjuntos de dados com mais da metade dos dados falhos,

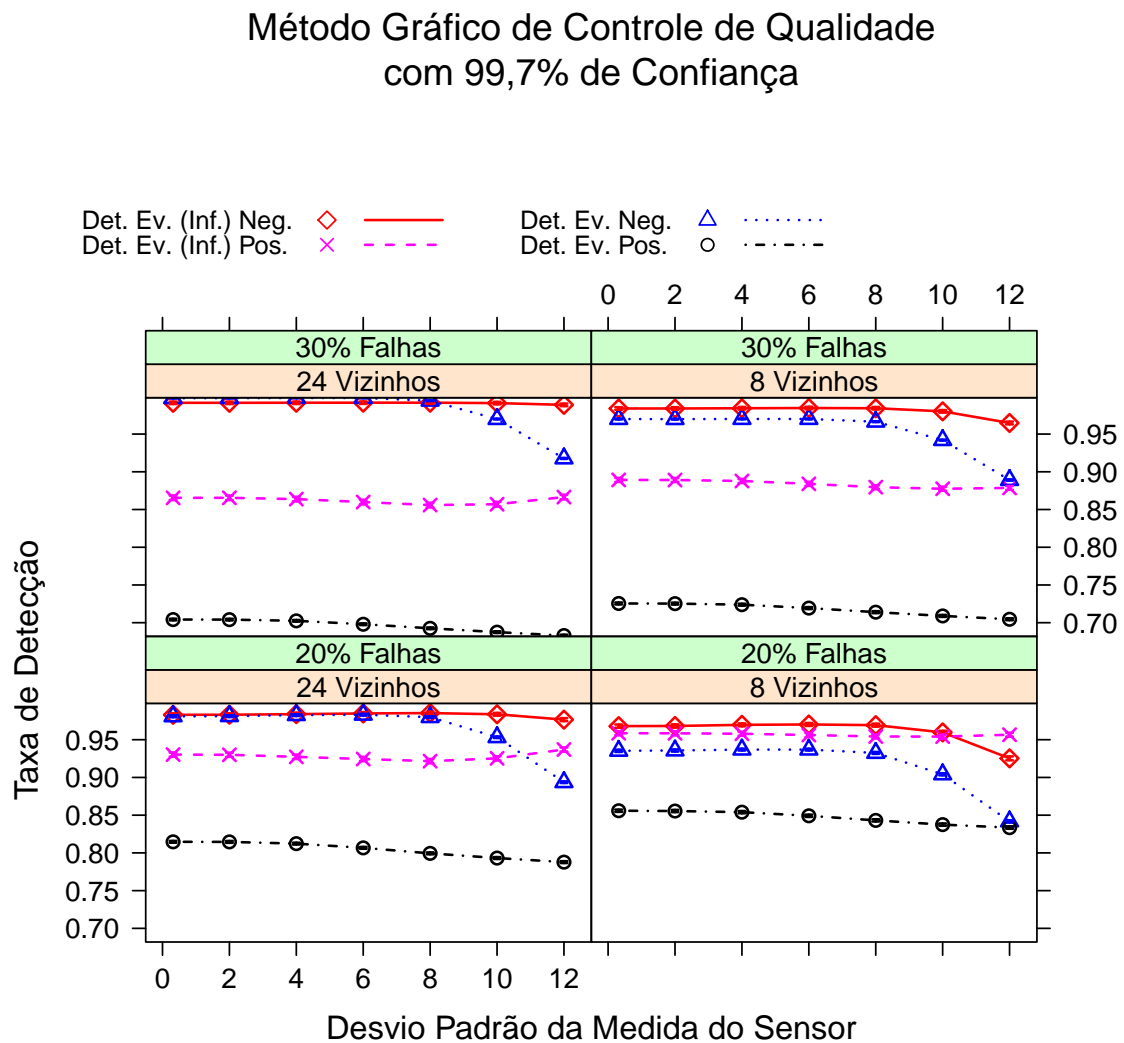


Figura 5.5: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, com 20% e 30% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

a taxa positiva de detecção de falhas cai drasticamente. Isto acontece porque nestes casos, a falha nunca será detectada.

Como a taxa negativa de detecção de falhas mantém-se alta, poucos nós que não falham substituem suas medidas pela média dos dados da sua vizinhança. Portanto, as falhas nas detecções se concentram nos nós falhos que não foram identificados. Como

Método Gráfico de Controle de Qualidade  
com Confiança de 99,7% (Desvio Padrão 4)

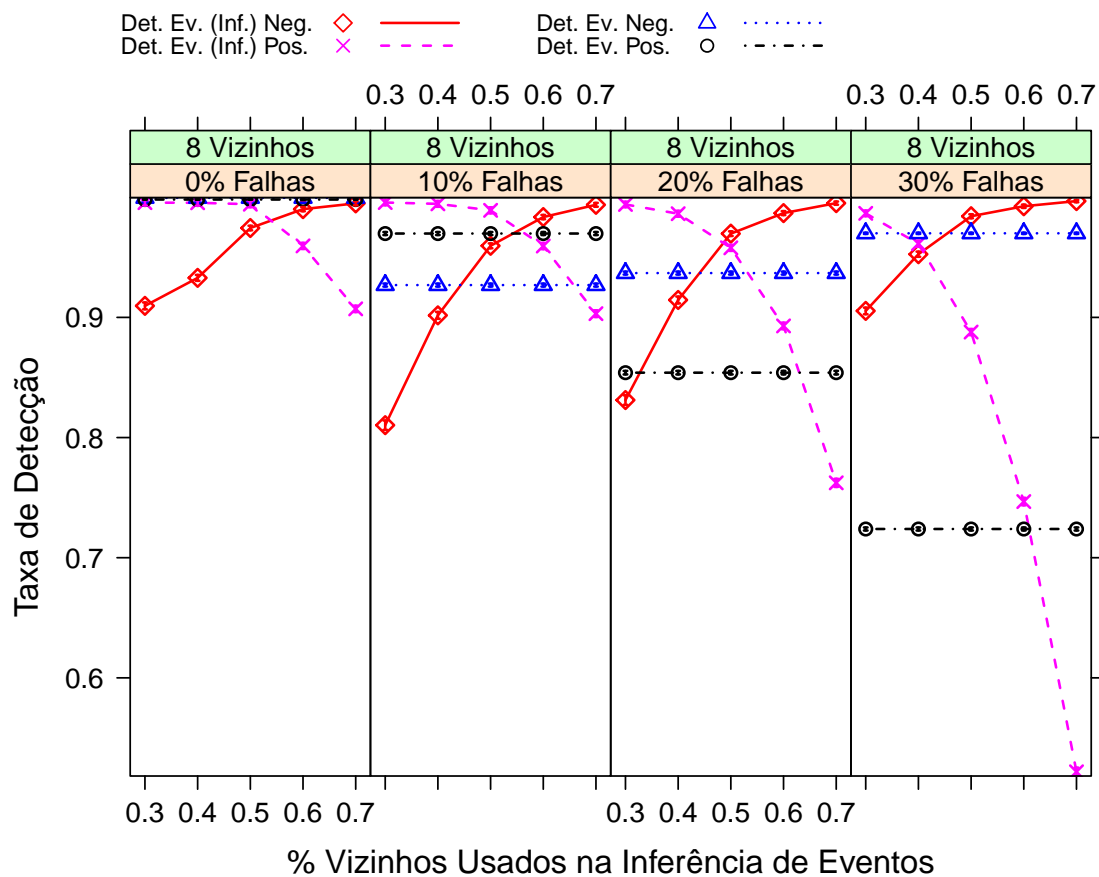


Figura 5.6: Taxa de detecção vs. porcentagem de vizinhos usados na inferência de eventos usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade e desvio padrão igual a quatro, com variação de falhas de nós.

as falhas são independentes espacialmente, a inferência de eventos consegue melhorar muito as taxas de detecção de eventos.

A melhora das taxas negativas é recorrente do impacto do efeito da piora das taxas positivas. Mais nós assumem que não estão na área de ocorrência de eventos por causa da maior quantidade de falhas no conjunto de dados. Assim, maior a probabilidade de, eventualmente, isto coincidir com a realidade, já que se tem somente duas opções

de estados na rede.

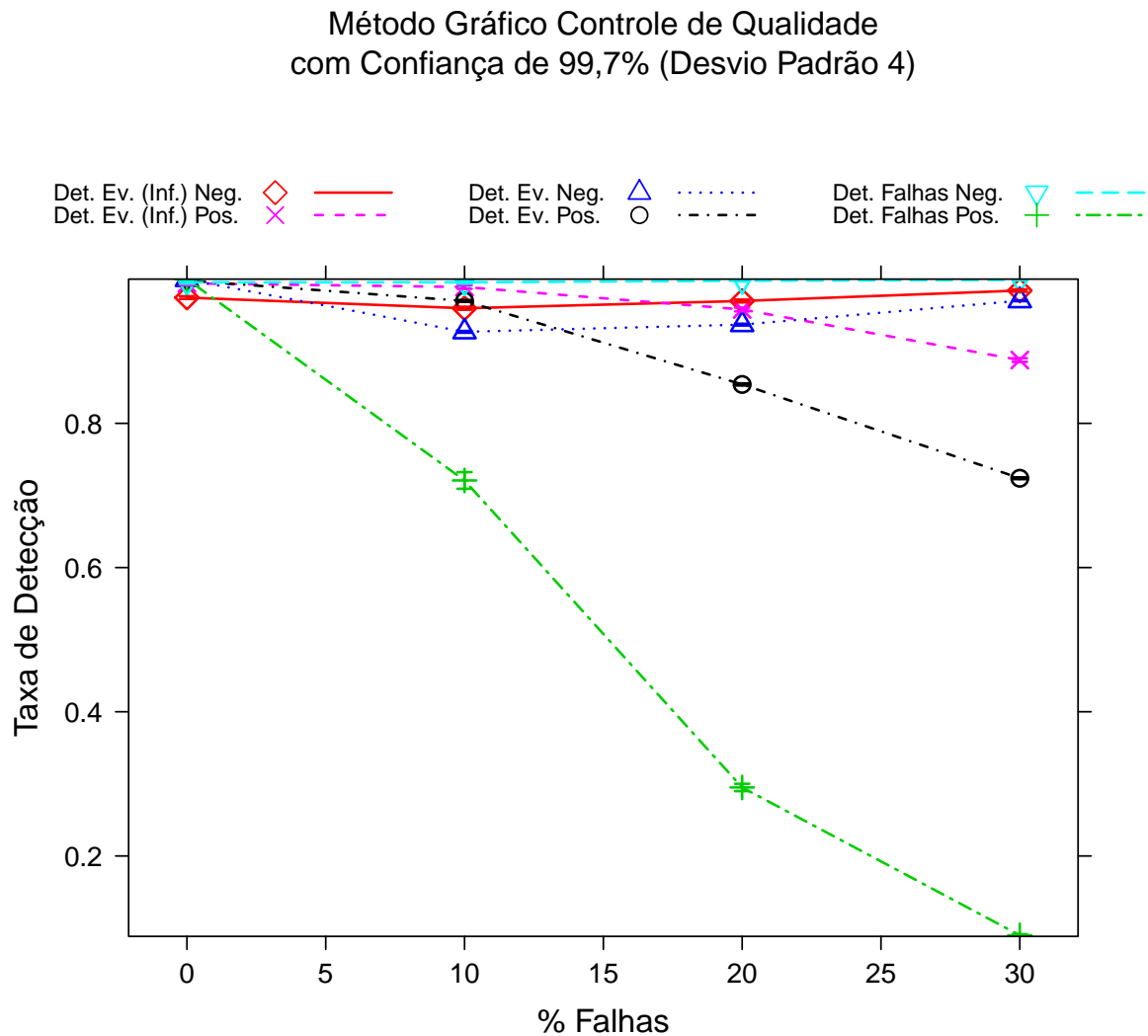


Figura 5.7: Taxa de detecção vs. porcentagem de falhas de sensores usando o Método Gráfico de Controle de Qualidade com 99,7% de confiança em um cenário em que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade e desvio padrão igual a 4.

### 5.3.2 Método Intervalo de Confiança

O método é baseado em intervalos de confiança para o cálculo do intervalo de referência de ocorrência de falhas. Este mesmo intervalo substituirá a medida do sensor

caso ela seja identificada como uma falha. Este método foi simulado para duas variações – 95% e 99,5% de intervalo de confiança.

### 5.3.2.1 Quantidade de Vizinhos, Desvio Padrão da Medida e Porcentagem de Falhas

A figura 5.8 apresenta o desempenho das taxas de detecção e de inferência quando se varia o  $\sigma$  da medida e a porcentagem de nós da rede que falham para o método Intervalo de Confiança com 95% de confiança. Com o aumento da porcentagem de nós falhos e do  $\sigma$  da medida, o desempenho das taxas negativas cai. Ou seja, com o aumento dos fatores que ampliam o tamanho dos intervalos envolvidos nas detecções, percebe-se uma queda no desempenho das taxas negativas e uma melhora no das taxas positivas.

Ainda de acordo com o gráfico, pode-se perceber que o método Intervalo de Confiança é mais sensível a falhas que o método Gráfico de Controle de Qualidade, para esta configuração. Isto acontece porque todo o conjunto de dados recebido pelo sensor da sua vizinhança é utilizado para fazer o intervalo de confiança usado como referência pelo método.

Mesmo para 0% de ocorrência de falhas de sensores, a taxa de detecção negativa de falha é baixa para valores pequenos de  $\sigma$ . Com isto, conclui-se que os intervalos comparados não são grandes o suficiente para se intersectarem quando o  $\sigma$  da medida é pequeno. Aumentando a taxa de falhas de sensores, aumenta-se a quantidade de falhas no conjunto de dados usado para calcular o intervalo de confiança. Assim a média do intervalo é deslocada em direção ao intervalo que caracteriza falhas.

Quanto maior a quantidade de falhas no conjunto de dados, mais próxima do intervalo que caracteriza a ocorrência de falhas estará a média do intervalo de referência. Então, as medidas falhas serão detectadas como valores válidos e as medidas válidas como falhas. Com o aumento do  $\sigma$  da medida, este quadro se inverte, pois os intervalos variam mais de valores que os gerados com baixo desvio padrão, aumentando as chances da detecção correta.

A taxa de detecção de falhas é elevada mesmo para 30% de falhas de sensores, quando temos valores elevados de  $\sigma$ . O impacto desta detecção é substituir as medidas consideradas falhas pelo intervalo de confiança dado pelas medidas dos vizinhos, melhorando a qualidade das taxas de detecção e inferência de eventos.

Pela escolha dos valores de medidas com e sem falhas e do intervalo considerado como evento, as falhas não detectadas pioram o desempenho das taxas negativas. As medidas com falha afastam a média do intervalo de confiança do valor normal e



a aproximam do intervalo que caracteriza a ocorrência de eventos, aumentando a ocorrência de falsos positivos.

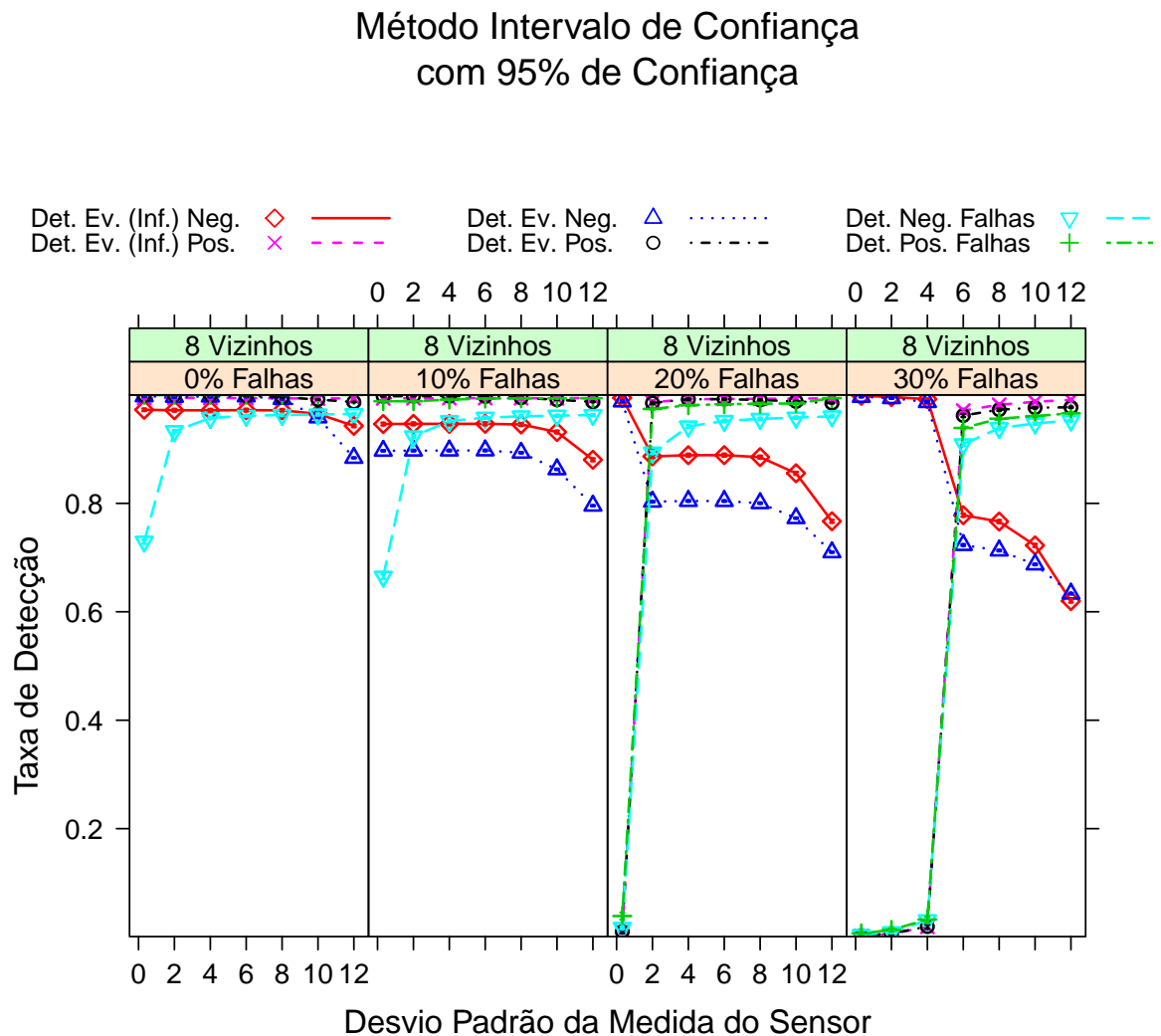


Figura 5.8: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Intervalo de Confiança com 95% de confiança em um cenário em que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade, variando-se as falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

A figura 5.9 apresenta os resultados das simulações com 0% e 10% de falhas de sensores nos cenários em que se recebe dados de oito e 24 vizinhos, variando-se o  $\sigma$  da medida do sensor. Os resultados mostram que a configuração do Método Intervalo de

Confiança com 95% de confiança, com as porcentagens de falhas de sensores citadas, é mais estável para um conjunto de oito vizinhos. Isto porque quanto maior o conjunto de dados fornecidos pelos vizinhos, maior a probabilidade de ocorrência de medidas falhas no conjunto de dados.

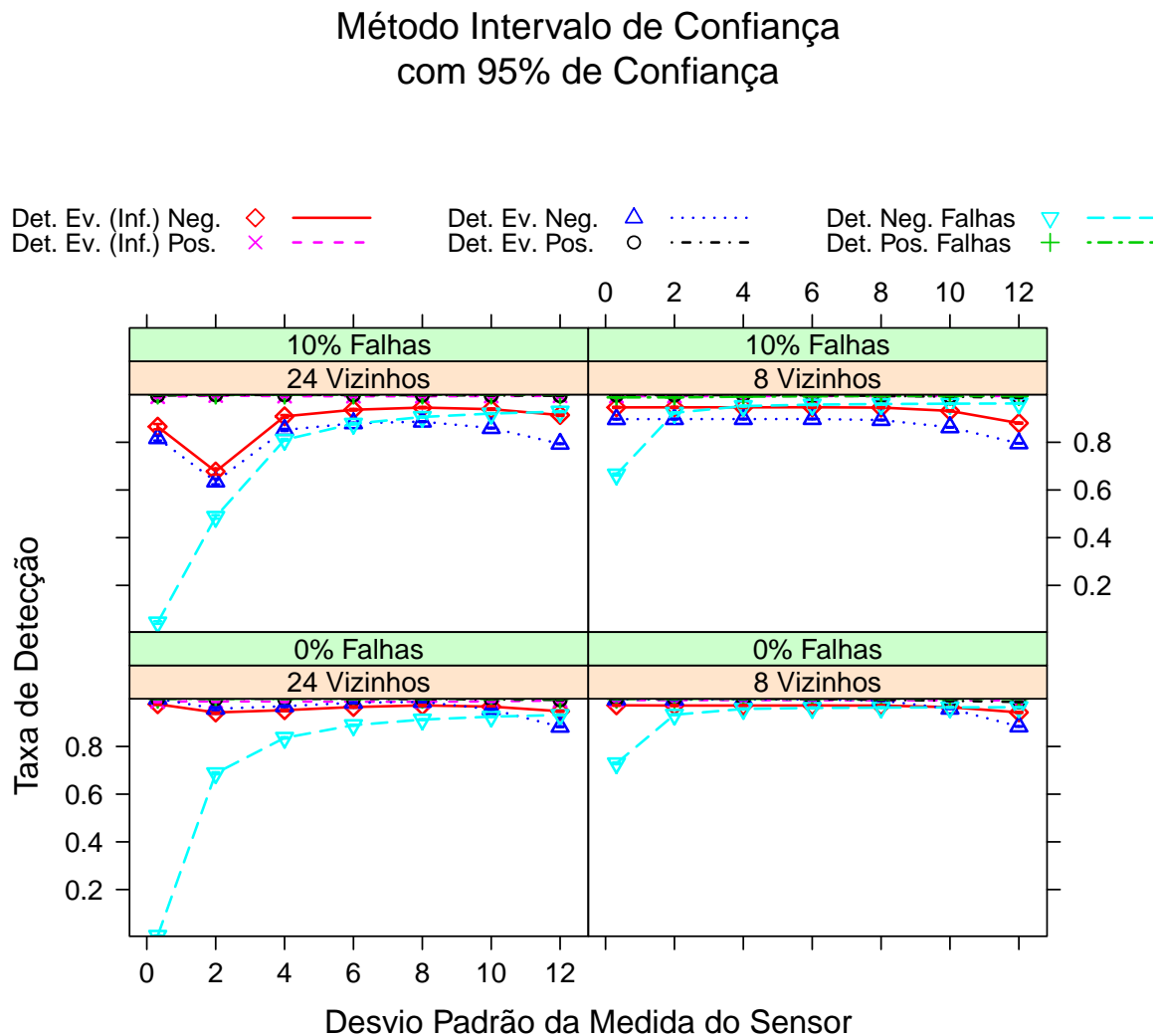


Figura 5.9: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Intervalo de Confiança com 95% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, para 0% e 10% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

A figura 5.10 apresenta os resultados para 20% e 30% de falhas de sensores para

oito e 24 vizinhos que confirmam as observações feitas para o Gráfico 5.9. Com o aumento da porcentagem de falhas, tem-se que, progressivamente, o desempenho da simulação que usa o conjunto de dados fornecidos por 24 vizinhos é pior do que para a que usa o conjunto de dados fornecidos por oito vizinhos.

Percebe-se um comportamento padrão que se repete: o aumento das taxas positivas com a diminuição das taxas negativas, com a exceção da taxa de detecção de falhas. Como as detecções acontecem em sequência, a detecção de falhas é a única que não sofre impacto de outras etapas do processo.

Observando que os resultados apresentados pelo método melhoravam com o aumento do tamanho dos intervalos envolvidos, foram feitas simulações com a variação 99,5% de intervalo de confiança. As figuras 5.11 e 5.12 apresentam os resultados para 0%, 10%, 20% e 30% de falhas de sensores usando oito e 24 vizinhos. O método, para o cenário usando os dados de oito vizinhos, apresenta resultados estáveis para todos os valores de desvio padrão testados. Pelos mesmos motivos já apresentados, as simulações que usaram 24 vizinhos mostram um comportamento menos estável para valores baixos de  $\sigma$ . Com o aumento da porcentagem das falhas de sensores, percebe-se uma queda acentuada da taxa de detecção de falhas positivas, mas ainda assim, visivelmente melhor que a detecção realizada pelo método anterior.

A figura 5.13 apresenta, em detalhes, os gráficos anteriores para as simulações com os oito vizinhos mais próximos. Pode-se perceber as progressivas quedas das taxas positivas e negativas com o aumento da porcentagem de nós falhos. Como a taxa de detecção de falhas cai, mais nós sensores assumem que suas medidas falhas são valores verdadeiros. Isto reflete nas outras taxas de detecção: tanto as taxas positivas quanto negativas caem. Como a queda de uma taxa pode aumentar o valor da outra taxa, percebe-se que o valor acrescido pela queda de desempenho da taxa complementar (no caso da taxa positiva é a taxa negativa e vice-versa) não compensa a piora na detecção de cada taxa. Este quadro estabiliza para taxa negativa quando se tem 30% de falhas de sensores.

### **Porcentagem de Vizinhos Usada no Método de Inferência de Eventos**

Esta seção apresenta as simulações para a versão do método com 99,5% de confiança. A figura 5.14 apresenta o comportamento do método de inferência usado em conjunto com o método Intervalo de Confiança para detecção de falhas e eventos. A simulação usou o conjunto de dados formado pelas medidas com  $\sigma$  igual a 4 fornecidas pelos oito vizinhos mais próximos. Na figura, as detecções positiva e negativa de eventos aparecem apenas como referências. O cenário padrão de simulação deste método

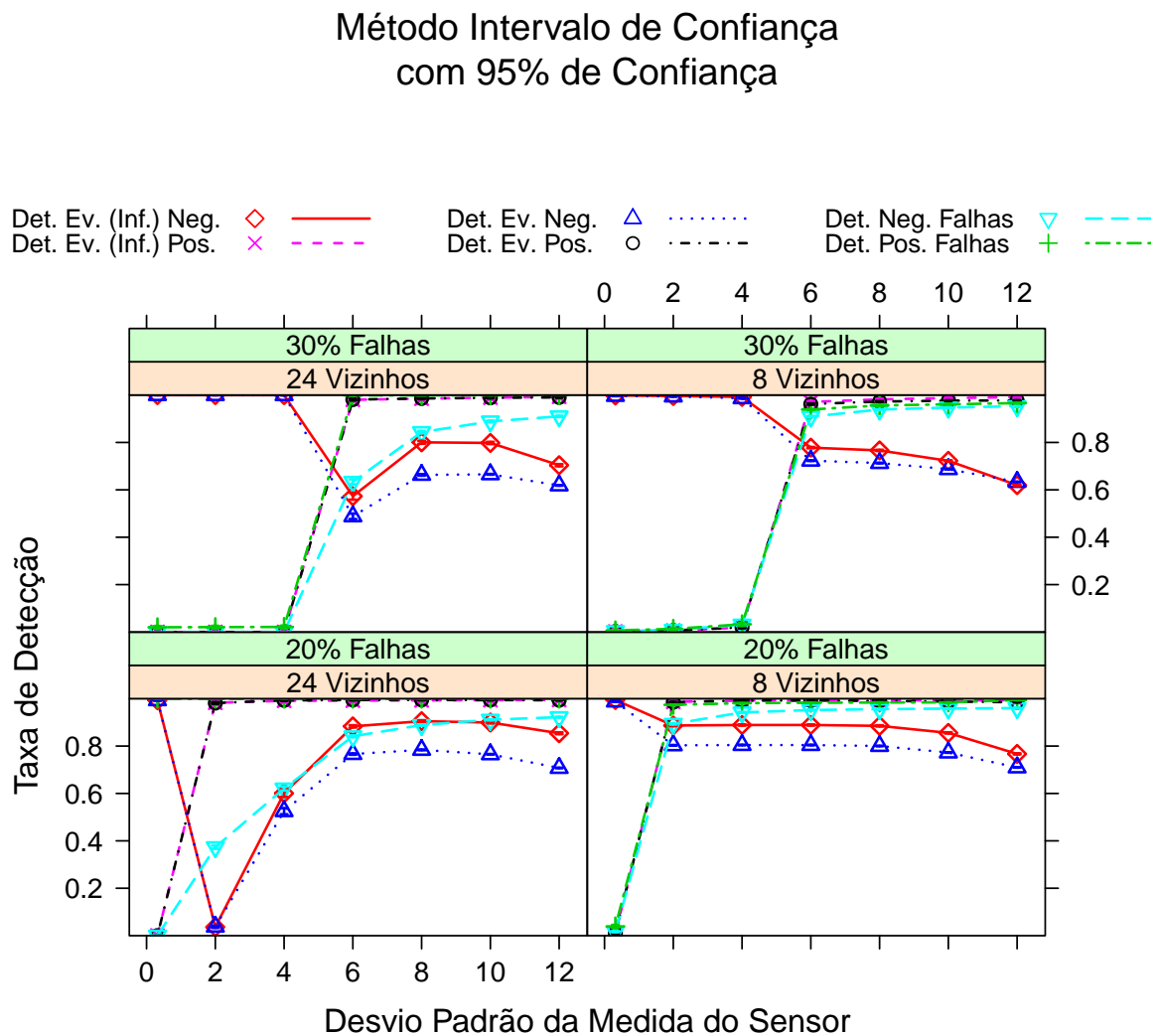


Figura 5.10: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Intervalo de Confiança com 95% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, para 20% e 30% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

usa 50% dos vizinhos como limiar de inferência de eventos.

A figura mostra o comportamento das taxas positiva e negativa de inferência de eventos para o método. Tal como o método Gráfico de Controle de Qualidade, quanto maior a porcentagem de vizinhos usados na inferência melhor é a taxa negativa pelas mesmas justificativas.

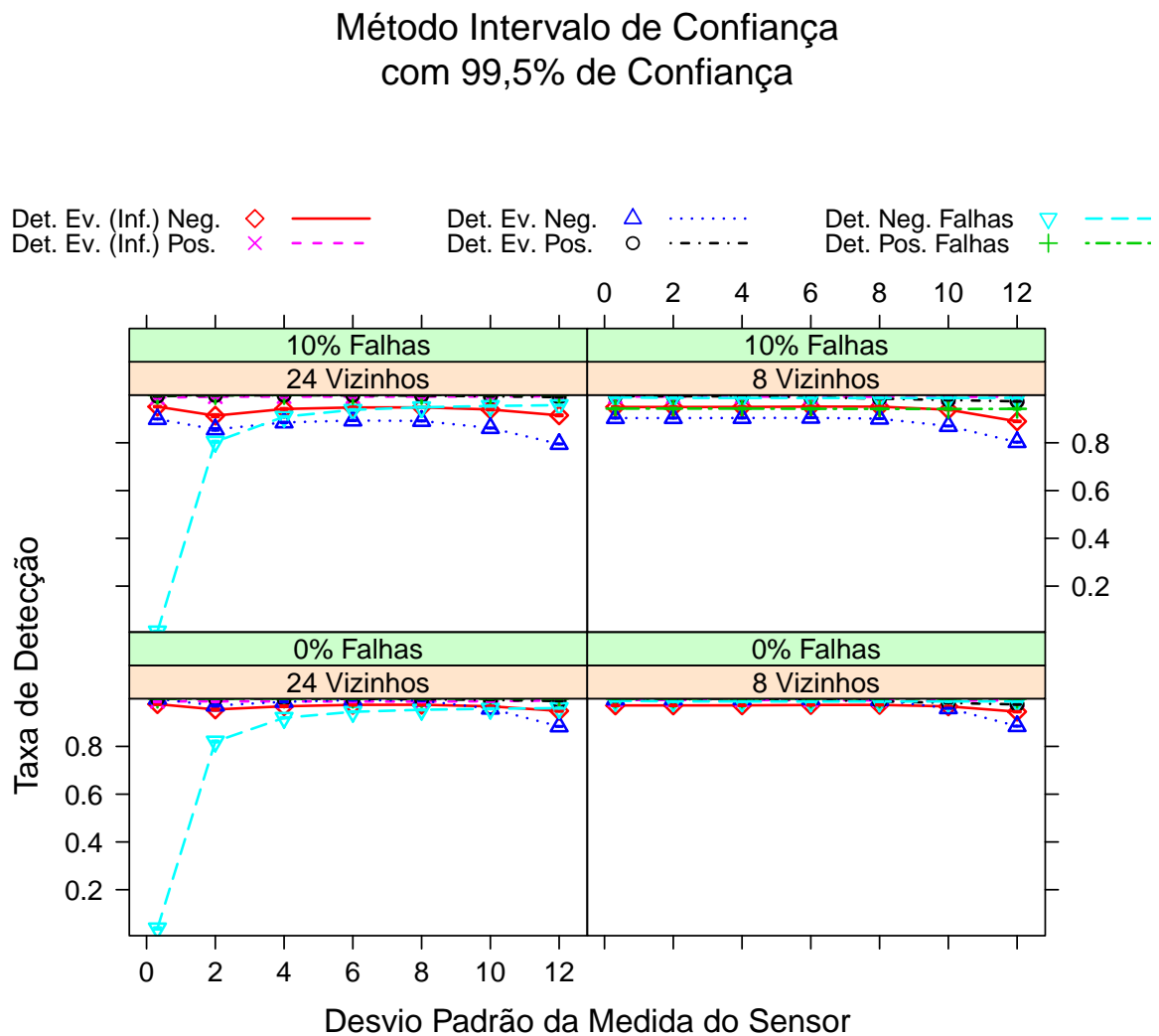


Figura 5.11: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Intervalo de Confiança com 99,5% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, para 0% e 10% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

O Gráfico 5.15 sumariza o efeito das taxas positivas e negativas, previamente detalhadas, variando-se a porcentagem de falhas de nós sensores. Observa-se que as taxas positivas caem progressivamente com o aumento da porcentagem de falhas. Já as taxas negativas, apesar de também apresentarem quedas com o aumento da porcentagem, apresentam quedas menos acentuadas e uma tendência a se estabilizarem quando as

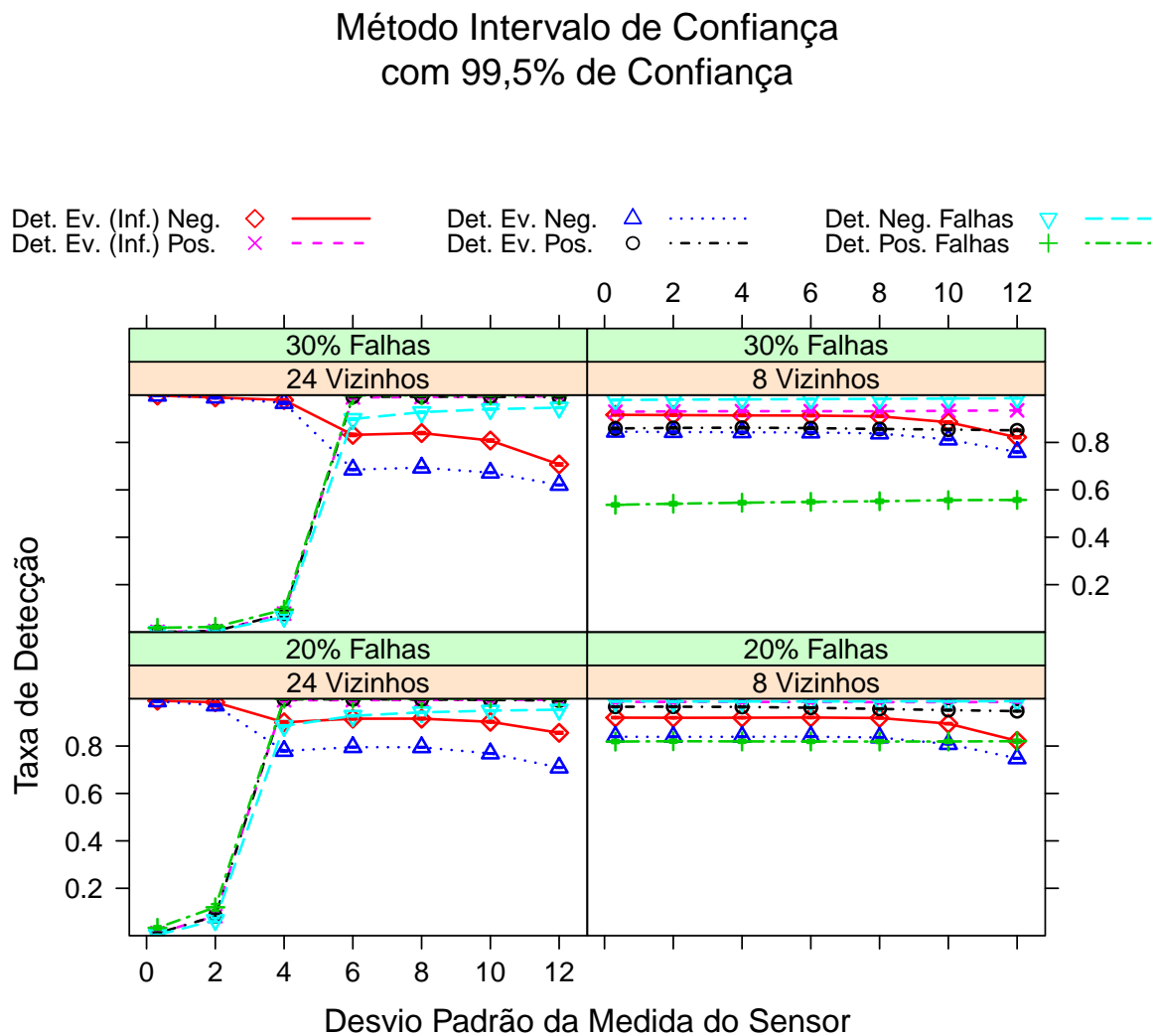


Figura 5.12: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Intervalo de Confiança com 99,5% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, para 20% e 30% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

porcentagens de nós falhos aumentam.

As taxas negativas, cujos acertos são caracterizados pela não intersecção dos intervalos envolvidos, caem por causa do aumento do tamanho dos intervalos quando aumenta a quantidade de falhas no conjunto de dados. Como o tamanho do conjunto é fixo, o aumento do intervalo causado por uma maior quantidade de falhas é limitado.

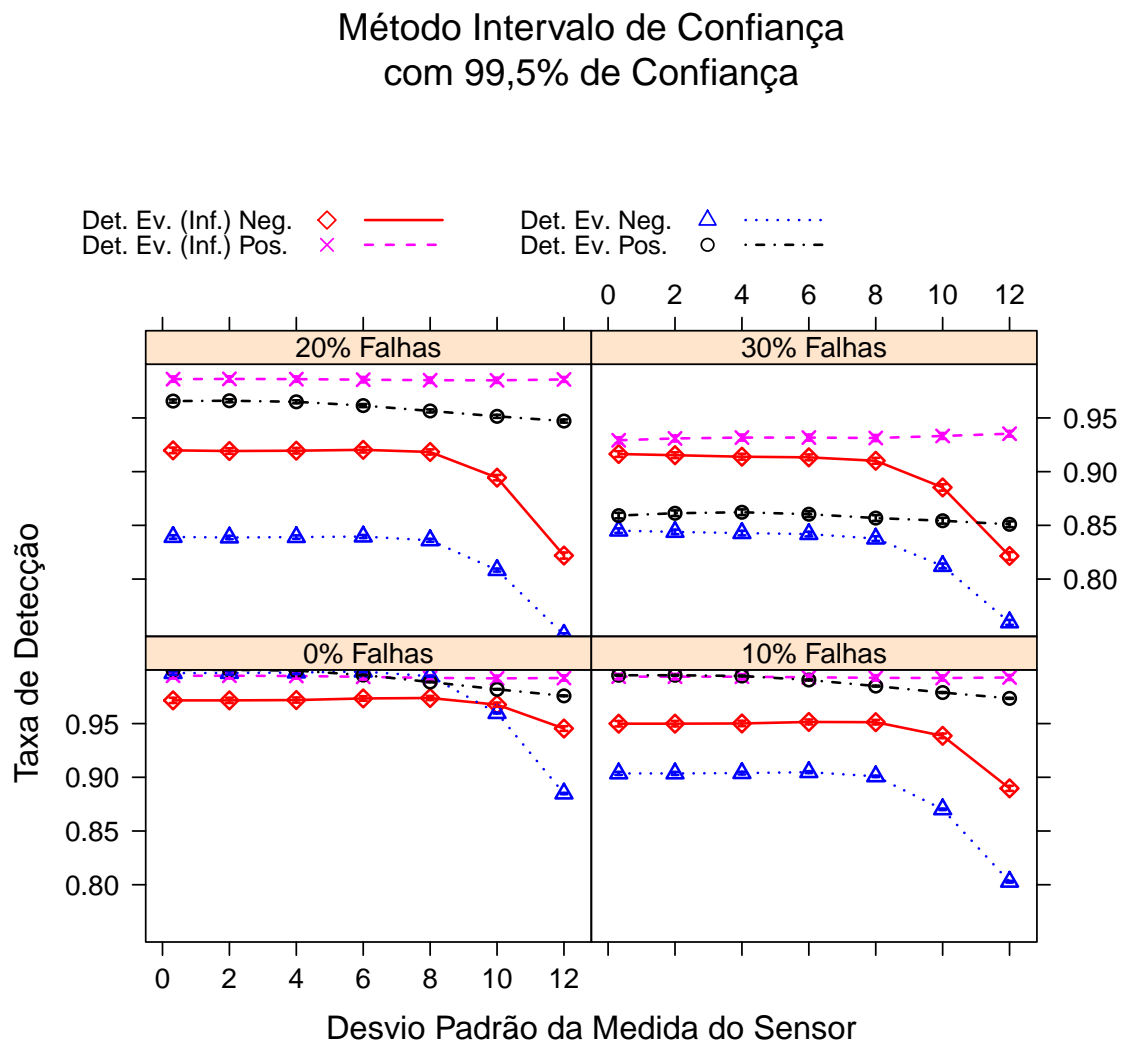


Figura 5.13: Em detalhes: Taxa de detecção vs. desvio padrão da medida dos sensores usando o Método Intervalo de Confiança com 99,5% de confiança em um cenário em que cada sensor recebe dados dos seus oito e 24 vizinhos mais próximos na grade, para 20% e 30% de falhas de nós. O Método de Inferência usa o limiar de 50% dos vizinhos com detecção positiva para inferir a ocorrência de um evento no nó.

Quando mais da metade dos dados do conjunto for de falhas, a tendência é a diminuição do tamanho do intervalo e, portanto, da melhora das taxas negativas, como visto no gráfico 5.15

Já as taxas positivas são caracterizadas pelas intersecção dos intervalos. Com o aumento da quantidade de falhas no conjunto de dados, ao mesmo tempo que o intervalo

Método Intervalo de Confiança  
com Confiança de 99,7% (Desvio Padrão 4)

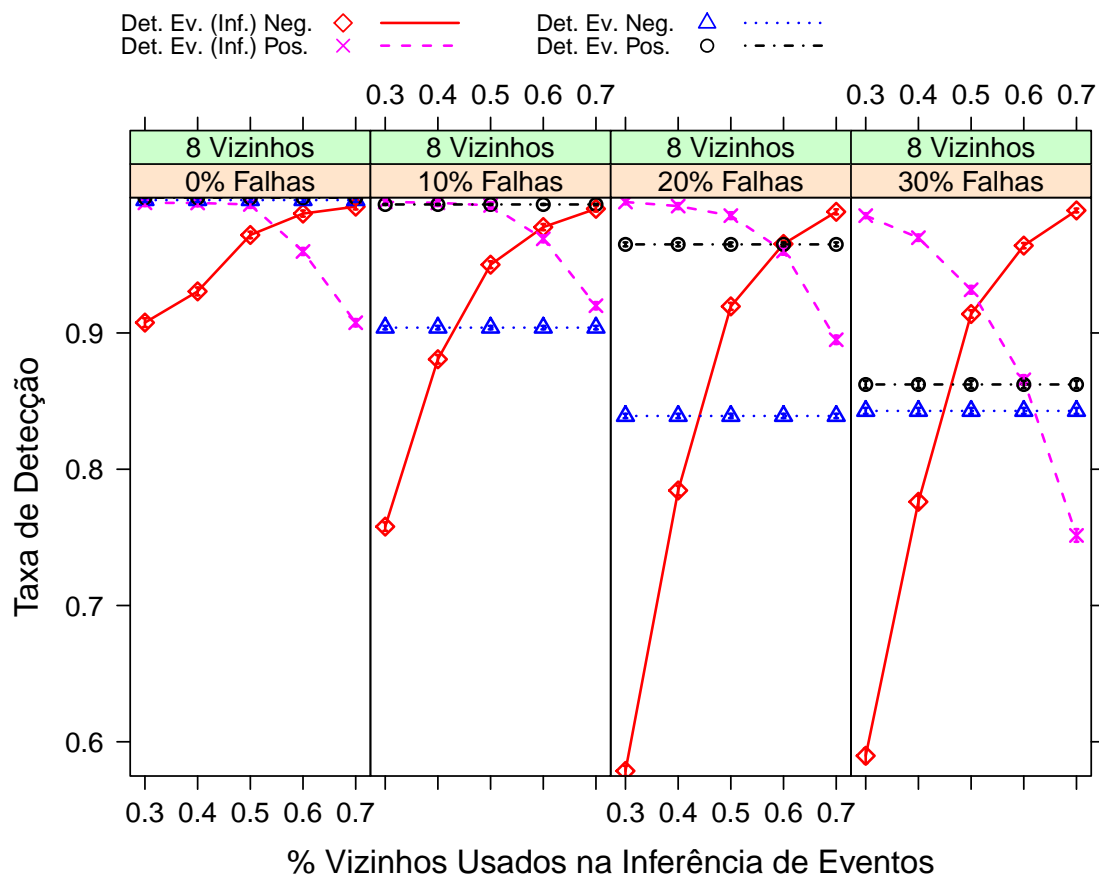


Figura 5.14: Taxa de detecção vs. porcentagem de vizinhos usados na inferência de eventos usando o Método Intervalo de Confiança com 99,5% de confiança em um cenário em que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade, variando-se a porcentagem de falhas de nós.

aumenta, ele desloca-se na direção do valor da falha. A queda inicial da taxa é causada por este deslocamento. Quando a quantidade de falhas cresce o suficiente para o intervalo começar a diminuir, a queda das taxas positivas torna-se mais acentuada.



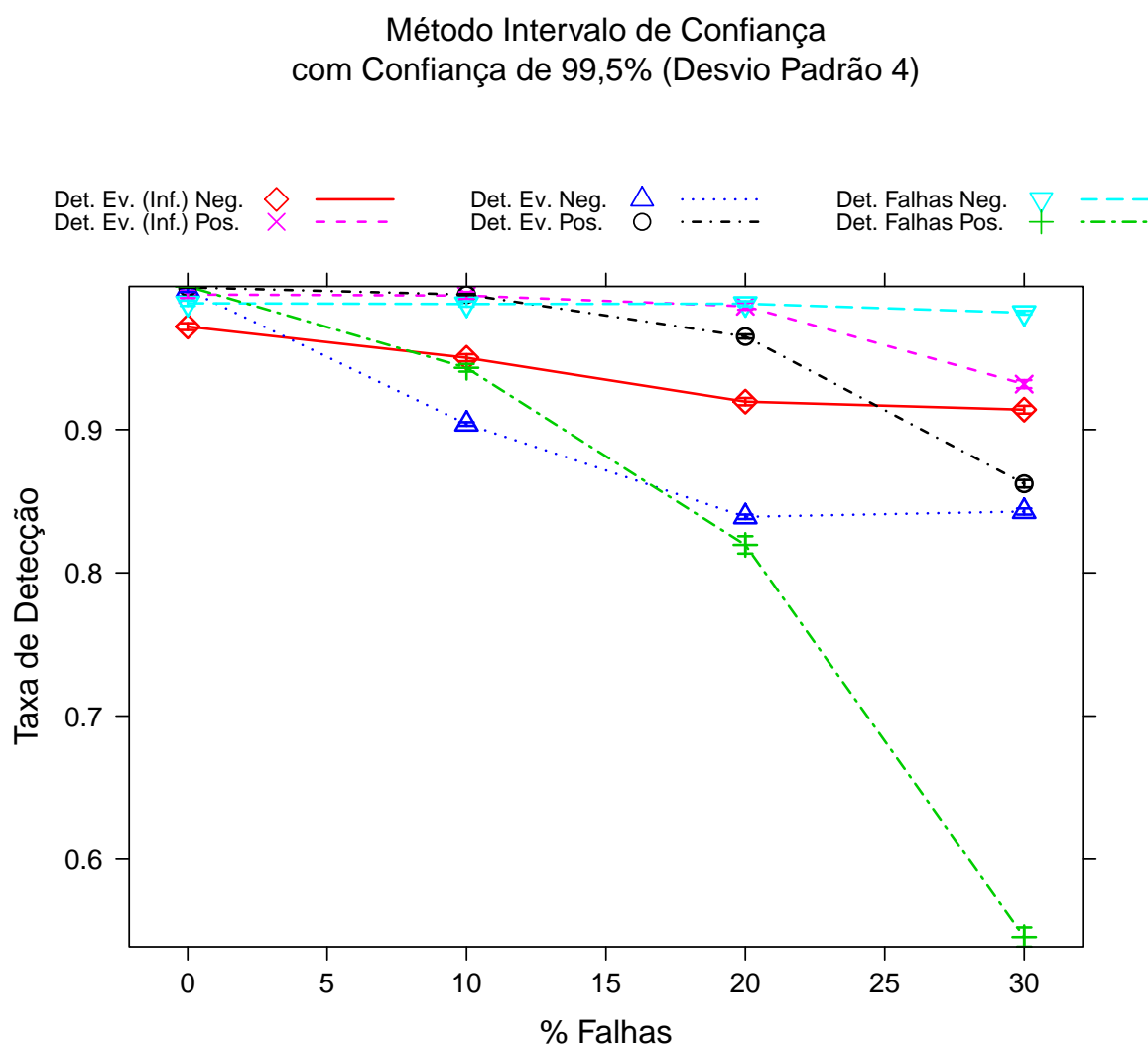


Figura 5.15: Taxa de detecção vs. porcentagem de falhas de nós sensores usando o Método Intervalo de Confiança com 99,5% de confiança em um cenário em que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade, com desvio padrão igual a 4.

### Comparação dos Métodos

Os Gráficos 5.16 e 5.17 comparam os métodos apresentados em relação às taxas positivas e negativas, respectivamente. As taxas apresentadas nos gráficos foram previamente descritas nas seções anteriores. Foram comparados os métodos Gráfico de Controle de Qualidade com 99,7% de confiança e Intervalo de Confiança com 99,5% de

confiança.

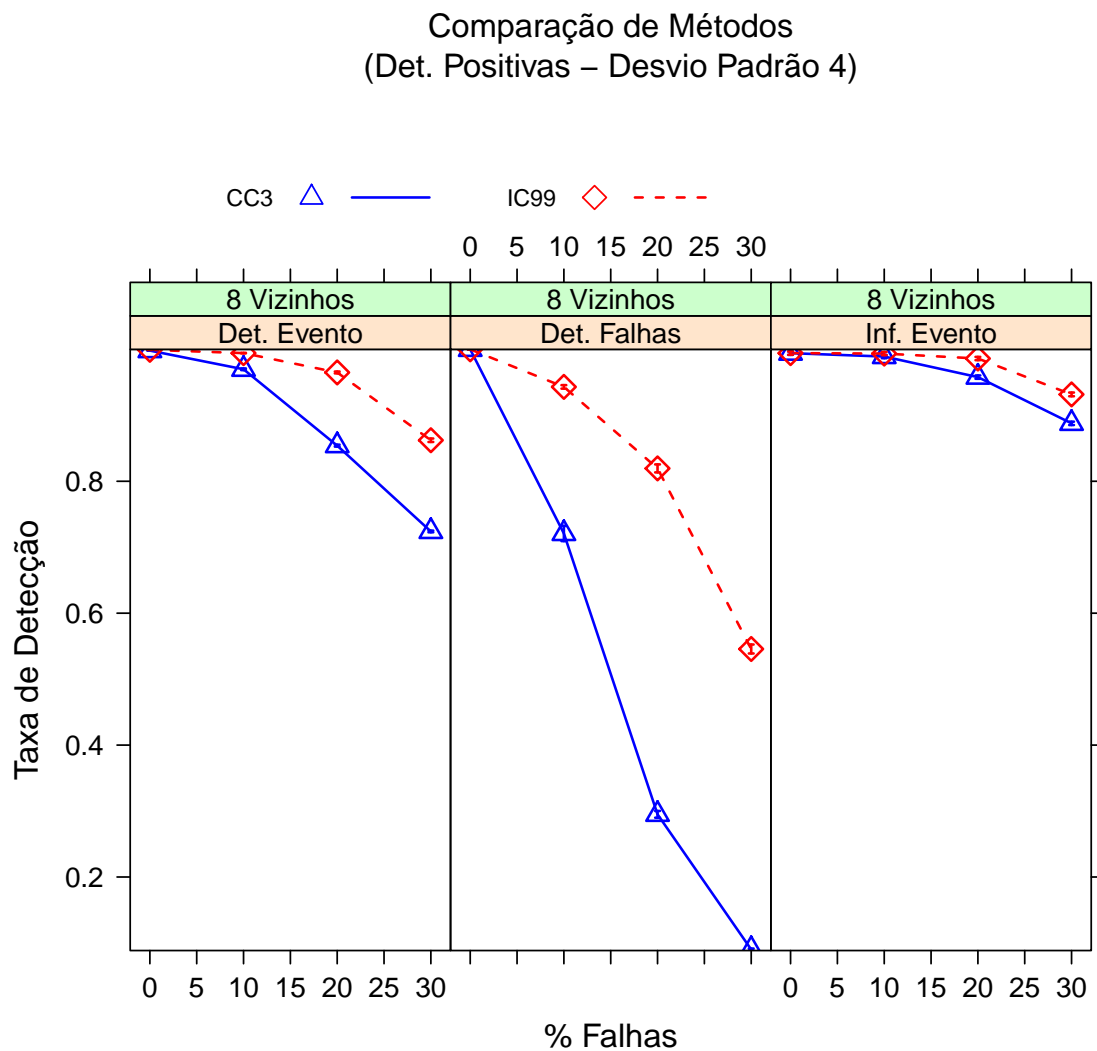


Figura 5.16: Comparação das Taxa de detecção positivas dos métodos vs. porcentagem de falhas de sensores em um cenários que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade e com medidas de sensores com desvio padrão igual a quatro.

O Gráfico 5.16 compara as taxas positivas dos métodos propostos. Como se pode observar, o método Intervalo de Confiança apresentam melhores resultados para este tipo de taxa. Apesar da maior diferença entre os métodos em relação às taxas positivas de detecção de falhas e eventos, a inferência de eventos possui um resultado similar entre os métodos, cerca de 4% de diferença para 30% de falhas de nós sensores.

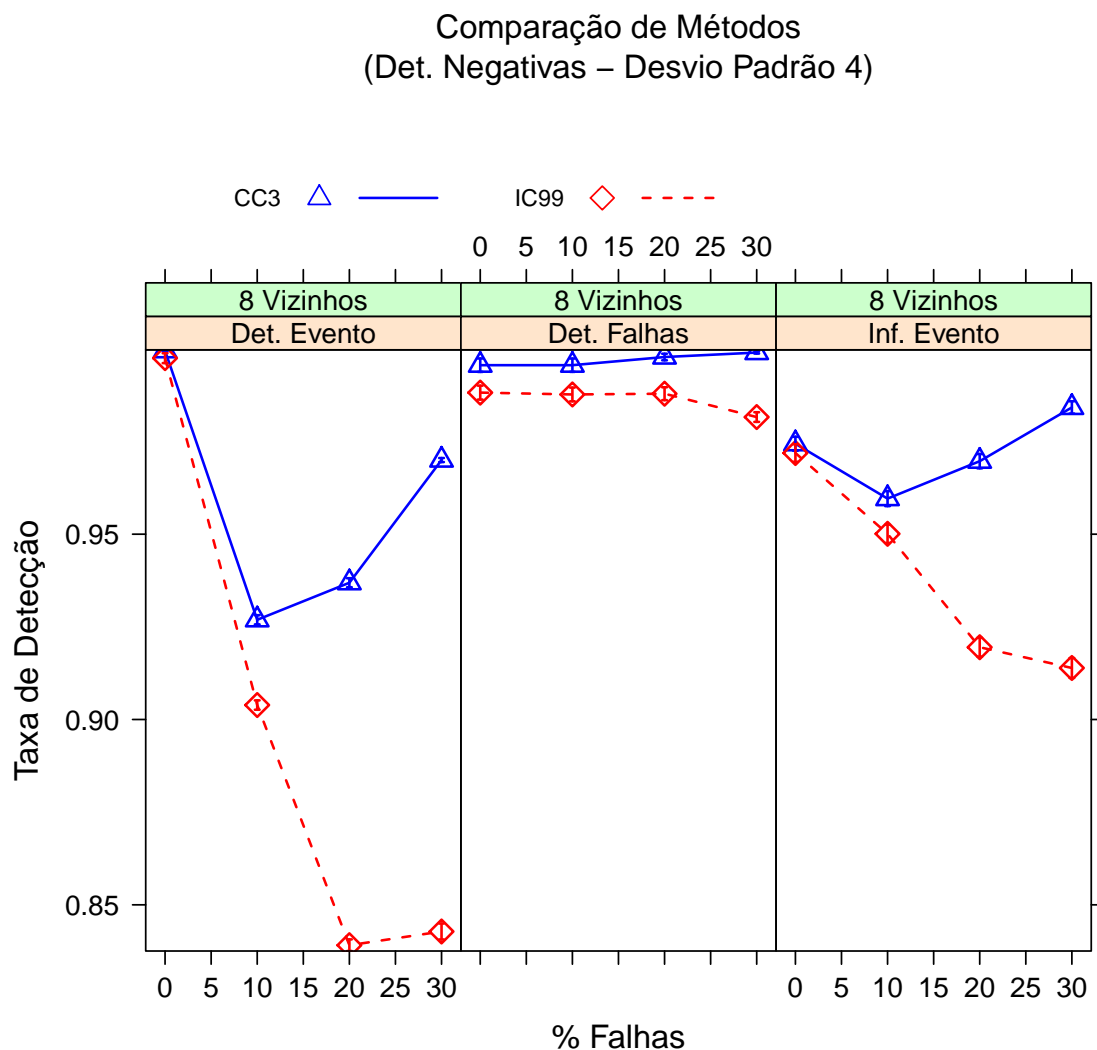


Figura 5.17: Comparação das Taxa de detecção negativas dos métodos vs. porcentagem de falhas de sensores em um cenários que cada sensor recebe dados dos seus oito vizinhos mais próximos na grade e com medidas de sensores com desvio padrão igual a quatro.

O Gráfico 5.17 compara as taxas negativas dos métodos propostos. Neste caso, o método Gráfico de Controle de Qualidade apresenta resultados superiores aos apresentados pelo método Intervalo de Confiança. Para 30% de falhas de nós sensores, o método Gráfico de Controle é cerca de 7% melhor que o método Intervalo de Confiança.

## 5.4 Considerações Finais

Este capítulo apresentou a metodologia usada e os resultados obtidos das simulações dos métodos propostos usando o arcabouço adaptado *Diffuse*. Para testar o arcabouço, propõe-se dois grupos de métodos: (i) métodos de detecção de falhas e eventos e (ii) um método de inferência de eventos. Em (i), tem-se os métodos Gráfico de Controle de Qualidade e o de Intervalo de Confiança. O primeiro é baseado na teoria de Controle Estatístico de Processo e o segundo em Intervalo de Confiança. Já em (ii), tem-se um método de inferência baseado em corte por limiar.

Os métodos foram analisados segundo as métricas de taxas positivas e negativas de detecção de falhas e eventos e inferência de eventos. Taxa positiva é a taxa de verdadeiro positivo e taxa negativa é a de verdadeiro negativo. Variando-se  $\sigma$  das medidas realizadas pelos nós sensores e porcentagens de nós falhos na rede, verifica-se a taxas de acertos positivos e negativos dos métodos.

No primeiro conjunto de simulação, o arcabouço foi configurado com um filtro de média móvel e o método Gráfico de Controle de Qualidade com 99,7% como detector de falhas de nós sensores, o mesmo método proposto para detecção de eventos e um método de inferência de eventos baseado em corte por limiar. Os resultados destas simulações mostraram uma detecção de eventos com uma taxa positiva de 96.98% e uma taxa negativa de 92.69% em um cenário de 10% de falhas de sensores,  $\sigma$  igual a quatro e conjunto de dados formado pelas coletas dos oito vizinhos mais próximos. No mesmo cenário, as taxas da inferência de eventos foram de 98.91% e 95.96%, para as taxas positivas e negativas, respectivamente, com um limiar de 50% para o método de inferência. Com 30% de falhas de sensores, no mesmo cenário, os resultados mostram uma detecção de eventos com uma taxa positiva de 72.39% e uma taxa negativa de 97.00%. Sendo que, para as taxas relativas à inferência com um limiar de 40%, tem-se uma taxa positiva de 96.13% e uma taxa negativa de 95.26%.

No segundo conjunto de simulação, o arcabouço foi testado usando um filtro de média móvel e o método Intervalo de Confiança com 99,5% como detector de falhas de nós sensores, o mesmo método para detecção de eventos e um método de inferência de eventos baseado em corte por limiar. Os resultados destas simulações mostraram uma detecção de eventos com uma taxa positiva de 99.43% e uma taxa negativa de 90.39% em um cenário de 10% de falhas de sensores,  $\sigma$  igual a quatro e conjunto de dados formado pelas coletas de oito vizinhos mais próximos. No mesmo cenário, as taxas de inferência de eventos foram de 99.36% e 95.01%, para as taxas positivas e negativas, respectivamente, com um limiar de 50% para o método de inferência. Com 30% de falhas de sensores, os resultados mostram uma detecção de eventos com uma

taxa positiva de 86.22% e uma taxa negativa de 84.28%. Sendo que, para o mesmo cenário para as taxas relativas à inferência, com um limiar de 50%, tem-se uma taxa positiva de 93.17% e uma taxa negativa de 91.39%. As altas taxas de acerto, tanto positivas quanto negativas, mesmo para 30% de falhas de sensores, sugerem que o arcabouço cumpriu seu propósito de detecção cobrindo tolerância a falhas e imprecisão de medidas.

Como se pode observar, o método de inferência de eventos melhora os resultados obtidos pela detecção de eventos. Isto acontece porque este método melhora a detecção na borda do evento, diminuindo, também, erros do tipo I e II, combinando correlação espacial e o resultado da detecção binária de eventos dos métodos propostos.

Os resultados mostram que os métodos propostos são indicados para cenários diferentes. Caso se tenha um cenário em que seja preciso uma detecção mais acurada de verdadeiro positivo, recomenda-se o uso do Método Intervalo de Confiança. Já no caso em que é necessária uma detecção mais precisa de verdadeiro negativo, recomenda-se o uso do Método Gráfico de Controle de Qualidade.

## Capítulo 6

# Conclusão e Trabalhos Futuros

Esse trabalho apresentou uma visão geral sobre diversas abordagens para o problema de detecção de eventos em RSSFs. As abordagens do problema seguem diferentes modelagens e metodologias que tornam a comparação entre os métodos muito custosa. Para suprir a necessidade de comparar trabalhos relacionados à detecção de eventos, apresentou-se uma modelagem sistemática para comparar trabalhos de detecção de evento, composta pelo arcabouço *Diffuse* estendido e um conjunto de modelos-padrão dos elementos do problema.

O cenário de detecção de evento considera a imprecisão de dados coletados e a ocorrência de falhas durante as medidas dos nós sensores. Esse cenário busca ser realista: instrumentos sensores não-ideais e ocorrência de falhas de medidas de causas desconhecidas.

Foram modelados os dispositivos sensores, as medidas feitas por eles, a ocorrência de falhas de medidas, a ocorrência de eventos e a sua detecção. Os sensores são dispositivos eletrônicos não-ideais com precisão limitada de medida. As medidas realizadas são intervalos numéricos formados pela observação da medida ideal, ruído do ambiente e precisão do dispositivo. Um sensor realiza uma medida falha quando a medida, por causa desconhecida, é inconsistente com as medidas feitas pela sua vizinhança. Inicialmente, sabe-se qual é o intervalo numérico que caracteriza a ocorrência de um evento. Caso não falhe, um nó sensor fará uma medida caracterizando a ocorrência de um evento se estiver dentro da região de ocorrência.

O processo de detecção de eventos estudado nesse trabalho baseia-se na sequência das etapas: a filtragem da medida do sensor, a detecção de falhas de medidas, a detecção de eventos e a inferência de eventos. Ocorre uma falha quando a medida feita por um sensor não intersecta o intervalo dado pela sumarização das medidas feitas pela vizinhança e, nesse caso, essa medida será substituída pela sumarização. Caso

contrário, a medida é válida e será usada na detecção de evento. Ocorre um evento quando a medida de um sensor intersecta o intervalo de referência dado de ocorrência de eventos. Já a inferência de evento acontece quando, pelo menos, uma determinada porcentagem de vizinhos de um sensor detecta a ocorrência de um evento.

O arcabouço estendido *Diffuse* é formado pelos seguintes componentes: (i) processador de medida, (ii) processador de característica, (iii) processador de evento, (iv) estimador de estado e (v) tomador de decisão. Em (i), a medida realizada pelo sensor é filtrada e, posteriormente é verificada a sua validade. Em (ii), a medida pré-processada é transformada em uma entidade que melhor identifica a ocorrência de evento que a característica monitorada. Em (iii), detecta-se a ocorrência de um evento, explorando-se as propriedades de similaridade espacial ou temporal dos dados. Em (iv), um refinamento da detecção é realizado, agora examinando as detecções de eventos realizadas pelos vizinhos. E por fim, em (v), notifica-se ao nó sorvedouro caso tenha sido detectado um evento. O arcabouço foi configurado com o filtro de média móvel e um método de detecção de falhas como processador de medidas, um método de detecção de eventos como processador de eventos e um método de inferência de eventos baseado em limiar (50% dos vizinhos) como estimador de estados. Por simplificação, não exploramos a etapa do processador de característica.

Na última etapa, referente à tomada de decisão, envia-se uma mensagem ao nó sorvedouro em caso afirmativo de inferência de evento. No entanto, as simulações terminaram na etapa de estimação do estado, pois o foco desse trabalho é o processo de detecção de evento por si só. Dessa forma, foram simulados os métodos Gráfico de Controle de Qualidade com 99,7% de confiança e Intervalo de Confiança com as variações de 95% e 99,5% de confiança, em que ambos métodos foram usados para detectar falhas e eventos nas mesmas simulações.

Os resultados mostraram as tendências dos métodos em relação às métricas. Foi medido o desempenho dos métodos em relação às taxas positivas (verdadeiro positivo) e às taxas negativas (verdadeiro falso) relacionadas à detecção e à inferência. O Método Intervalo de Confiança apresenta alta sensibilidade ao tamanho dos intervalos envolvidos. Fatores que aumentam o tamanho dos intervalos interferem fortemente nos resultados do método e ambas as taxas diminuem com o aumento da quantidade de falhas. As taxas negativas sofrem maior impacto com o aumento do  $\sigma$  da medida pois ele aumenta o tamanho dos intervalos envolvidos e conseqüentemente aumenta a probabilidade deles se intersectarem.

A maior estabilidade do Método Gráfico de Controle de Qualidade com o aumento da quantidade de falhas é por ele se basear em uma triagem prévia do conjunto de dados e da retirada das medidas falhas. Com o aumento da quantidade de falhas, o método

Gráfico de Controle de Qualidade não consegue detectá-las, podendo ser assumidas como dados verdadeiros. Com o aumento de falhas, esse método apresenta queda do desempenho das taxas positivas e melhora do desempenho das taxas negativas.

A última etapa do processo de detecção, a inferência de eventos, sempre melhora o desempenho das taxas de detecção de eventos para os dois métodos simulados. Isto acontece porque esse método melhora a detecção na borda do evento, diminuindo, também, erros do tipo I e II.

A comparação dos métodos mostra que cada um deles apresenta um cenário ideal de detecção. Para os casos em que se deseja taxas mais precisas de detecção positivas (verdadeiro positivo), o método Intervalo de Confiança é o mais recomendado. Já em um cenário que precisa de uma taxa mais precisa de detecção negativa (verdadeiro negativo), recomenda-se o uso do Método Gráfico de Controle de Qualidade.

## Trabalhos Futuros

Para trabalhos futuros, propõe-se a avaliação dos métodos propostos, mais detalhadamente, explorando outros tipos de falhas, tais como falhas de comunicação entre sensores e perdas de pacote. Outra proposta seria explorar outros modelos de ocorrência de evento, como os abordados em Nakamura & Souza (2010).

Outra extensão seria a detecção de eventos correlacionada temporalmente. Nesse trabalho, o estado atual de detecção não considera o histórico de detecção de eventos na região. Combinar a correlação espacial com a temporal poderia melhorar a qualidade da detecção.

Outro trabalho seria estender o arcabouço em mais dois módulos: (i) aprendizado de evento – descobrindo o que é considerado como sendo evento – e (ii) mapeamento do evento – detalhando o cenário da detecção durante a entrega da notificação ao nó sorvedouro. Em (i), o sistema de detecção deve identificar o que é um evento baseando-se no fato que um evento possui baixa frequência de ocorrência, utilizando, para tal, técnicas de aprendizado de máquinas, por exemplo. Em (ii), sabendo-se o que é um evento e usando técnicas estatísticas de sumarização como agregação de dados ou baseadas em funções de similaridade de dados, o sistema de detecção deve entregar um mapa evolutivo da ocorrência do evento para o nó sorvedouro.



# Referências Bibliográficas

- (1960). *The International Dictionary of Applied Mathematics*. Van Nostrand, Princeton.
- (1989). *Controle Estatístico do Processo*. Qualidade e produtividade do IMAM. IMAM, 2nd edição.
- Ai, C.; Hou, H.; Li, Y. & Beyah, R. (2009). Authentic delay bounded event detection in heterogeneous wireless sensor networks. *Ad Hoc Networks*, 7:599–613.
- Bahrepour, M.; Meratnia, N.; Poel, M.; Taghikhaki, Z. & Havinga, P. J. (2010). Distributed event detection in wireless sensor networks for disaster management. In *International Conference on Intelligent Networking and Collaborative Systems*.
- Banerjee, T.; Xie, B. & Agrawal, D. P. (2008). Fault tolerant multiple event detection in a wireless sensor network. *Journal of Parallel and Distributed Computing*, 68:1222–1234.
- Buratti, C.; Conti, A.; Dardari, D. & Verdone, R. (2009). An overview on wireless sensor networks technology and evolution. *Sensors*, 9:6869–6896.
- Camarão, P. C. B. (1989). *Glossário de Informática*. Livros Técnicos e Científicos Editora.
- Costa, A. F. B.; Epprecht, E. K. & Caroinetti, L. C. R. (2005). *Controle Estatístico de Qualidade*. Editora Atlas, São Paulo.
- de Holanda Ferreira, A. B. (1975). *Novo Dicionário da Língua Portuguesa*. Nova Fronteira, Rio de Janeiro.
- de Sousa, L. D.; Frery, A. C. & Loureiro, A. A. F. (2012a). Modelagem unificada de métodos de detecção de eventos em redes de sensores sem fio e aplicação em presença de dados anômalos. In *Congresso Nacional de Matemática Aplicada e Computacional*.

- de Sousa, L. D.; Frery, A. C.; Nakamura, E. F. & Loureiro, A. A. F. (2012b). Event detection framework for wireless sensor networks considering data anomaly. In *The Seventeenth IEEE Symposium on Computers and Communication*.
- Diebold, F. X. (2007). *Elements of Forecasting*. Thomson South-Western.
- Ding, M.; Chen, D.; Xing, K. & Cheng, X. (2005). Localized fault-tolerant event boundary detection in sensor networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies*.
- Eisberg, R. & Resnick, R. (1923). *Física Quântica: Átomo, Sólido e Partículas*. Editora Campus.
- Ekambaram, S. K. (1972). *Base Estatística dos Gráficos de Controle de Qualidade*. Editora Polígono, São Paulo.
- Etzion, O. & Niblett, P. (2010). *Event Processing in Action*. Manning Publications Co.
- Fragomeni, A. H. (1987). *Dicionário Enciclopédico de Informática*. São Paulo: Nobel - Rio de Janeiro: Campus. Publicado anteriormente em um volume.
- Gaura, E. & Newman, R. (2006). *Smart MEMS and Sensor Systems*. Imperial College Press.
- Grubbs, F. (1948). On estimating precision of measuring instruments and product variability. *Journal of the American Statistical Association*, 43:243–264.
- Grubbs, F. (1973). Errors of measurement, precision, accuracy and the statistical comparison of measuring instruments. *Tecnometrics*, 15:53–66.
- Gungor, V. G. & Akan, O. (2007). Delay aware reliable transport in wireless sensor networks. *International Journal of Communication Systems*, 20:1155–1177.
- Guralnik, D. B. (1980). *New World Dictionary of American Language*. Cleveland, Ohio.
- James, B. R. (2008). *Probabilidade: Um Curso em Nível Intermediário*. IMPA, Rio de Janeiro.
- Jeong, M.; Bae, J. G. & Koh, B. H. (2010). A feasibility study of damage tracking through the diffusive communication of wireless sensors. *International Journal of Precision Engineering and Manufacturing*, 11:23–29.

- Kannan, D. (1979). *An Introduction to Stochastic Processes*. North-Holland Scientific Publishers.
- Krishnamachari, B. & Iyengar, S. (2004). Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Transactions on Computers*, 53:241–250.
- Kruger, M. & Grosse, C. (2009). Event-based data acquisition in wireless sensor networks for structural health monitoring. *TM-Technisches Messen*, 76:568–577.
- Li, M.; Liu, Y. & Chen, L. (2008). Nonthreshold-based event detection for 3d environment monitoring in sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1699–1711.
- Lopes, N. S. (2004). Inferência no modelo de grubbs t-elíptico. Master's thesis, Universidade Estadual de Campinas.
- Luo, X.; Dong, M. & Huang, Y. (2006). On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55:58–70.
- Marco, G. D.; Yang, T.; Ikeda, M. & Barolli, L. (2007). Performance evaluation of wireless sensor networks for event-detection with shadowing-induced radio. *Mobile Information Systems*, 3:251–266.
- Nakamura, E. F. (2007). *Fusão de Dados em Redes de Sensores sem Fio*. PhD thesis, Universidade Federal de Minas Gerais.
- Nakamura, E. F.; Loureiro, A. A. F. & Frery, A. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39:725–743.
- Nakamura, E. F. & Souza, E. L. (2010). Towards a flexible event-detection model for wireless sensor networks. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pp. 459–462.
- Nittel, S. (2006). Ned: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks. In *7th International Conference on Mobile Data Management*.
- Oliveira, R. A. R. (2008). *Identificação de Eventos no Canal de Comunicação em Redes IEEE 802.11 Usando Wavelets e Controle Estatístico de Processo*. PhD thesis, Universidade Federal de Minas Gerais.

- Palpanas, T.; Papadopoulos, D.; Kalogeraki, V. & Gunopulos, D. (2003). Distributed deviation detection in sensor networks. In *ACM SIGMOD Record*.
- Pawlowski, A.; Guzman, J. L.; Rodríguez, F.; Berenguel, M.; Sánchez, J. & Dormido, S. (2009). Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control. *Sensors*, 9:232–252.
- Rajasegarar, S.; Leckie, C. & Palaniswami, M. (2008). Anomaly detection in wireless sensor networks. *IEEE Wireless Communications*, 15:34 – 40.
- Riehle, D. (1969). *Framework Design: A Role Modeling Approach*. PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH.
- Sinalgo (2011). Sinalgo - simulator for network algorithms.
- Sun, X. S. & Coyle, E. J. (2010). Low-complexity algorithms for event detection in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 28:1138–1148.
- Thuc, K.-X. & Insoo, K. (2011). A collaborative event detection scheme using fuzzy logic in clustered wireless sensor networks. *International Journal of Electronics and Communications*, 65:485–488.
- Tseng, Y.; Lin, T.; Liu, Y. & Lin, B. (2005). Event-driven messaging services over integrated cellular and wireless sensor networks: prototyping experiences of a visitor system. *IEEE Journal on Selected Areas in Communications*, 23:1133–1145.
- Wong, K. & Sayeed, A. (2002). Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine*, 19:17–29.
- Yim, S.-J. & Choi, Y.-H. (2010). An adaptive fault-tolerant event detection scheme for wireless sensor networks. *Sensors*, 10:2332–2347.
- Zoumboulakis, M. & Roussos, G. (2010). Complex event detection in extremely resource-constrained wireless sensor networks. *Mobile Networks and Applications*, 16:194–213.