

UNIVERSIDADE FEDERAL FLUMINENSE

Giulio Dariano Bottari

**Monitoramento Térmico Responsivo para Centros de  
Processamento de Dados**

NITERÓI

2014

UNIVERSIDADE FEDERAL FLUMINENSE

Giulio Dariano Bottari

# Monitoramento Térmico Responsivo para Centros de Processamento de Dados

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Redes e Sistemas Distribuídos e Paralelos.

Orientador:

Julius C. B. Leite, Ph.D.

Co-orientador:

Raphael Pereira de Oliveira Guerra, Dr.-Ing.

NITERÓI

2014

Giulio Dariano Bottari

Monitoramento Térmico Responsivo para Centros de Processamento de Dados

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Redes e Sistemas Distribuídos e Paralelos.

Aprovada em março de 2014.

BANCA EXAMINADORA

---

Prof. Julius Leite – Orientador, IC-UFF

---

Prof. Orlando Gomes Loques Filho, IC-UFF

---

Prof. Alexandre Sztajnberg, DICC-UERJ

Niterói

2014

# Resumo

A popularidade crescente de aplicações na Internet nos últimos anos vem impulsionando a expansão de *datacenters* ao redor do mundo. Essas instalações são responsáveis por 1,3% da energia elétrica consumida por todo o planeta, sendo que metade é gasta em distribuição de energia e arrefecimento. Dado que a energia que alimenta esses centros em grande parte é baseada em combustíveis fósseis, essa indústria é também altamente poluente. Ao mesmo tempo que procura-se economizar energia, é importante manter as aplicações do *datacenter* funcionando dentro de limites térmicos seguros. Para atingir esses objetivos, propõe-se um sistema de monitoramento térmico responsivo para *datacenters*. Além de monitorar em tempo real o ambiente através de uma rede de sensores sem-fio, o sistema é capaz de detectar automaticamente anomalias que podem estar relacionadas a problemas térmicos no *datacenter*. A ferramenta também é capaz de avisar com antecedência caso haja uma previsão de superaquecimento da instalação. O trabalho foi validado através de experimentos em um *datacenter* real.

**Palavras-chave:** Economia de energia, centros de processamento de dados, sistemas de monitoramento, redes de sensores sem-fio, detecção de anomalias em séries temporais, previsão térmica.

# Abstract

The increasing popularity of Internet applications in the last few years has been propelling datacenter expansion around the world. These facilities accounts for 1.3% of the electric energy consumed by the whole planet, considering that half is spent on power distribution and cooling. Given that the energy that feeds these centers is largely based on fossil fuels it is also a high-polluting industry. At the same time that it is sought to save energy, its important to keep these applications running within safe thermal boundaries. To achieve these goals, we propose a responsive thermal monitoring system for datacenters. Besides monitoring the environment in real-time through a wireless sensor network, the system is able to automatically detect anomalies that may be related to datacenter thermal issues. The tool is also capable of advance notice in case of overheating predictions for the facility. This work has been validated through experiments in a real datacenter.

**Keywords:** Energy saving, datacenters, monitoring systems, wireless sensor networks, time-series anomaly detection, thermal prediction.

# Sumário

<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Ambiente de um CPD . . . . .	2
1.2 Problemas térmicos comuns em CPDs . . . . .	3
1.3 Proposta . . . . .	5
1.4 Divisão de capítulos . . . . .	5
<b>2 Trabalhos relacionados</b>	<b>7</b>
2.1 Dinâmica dos Fluidos Computacional . . . . .	7
2.2 Monitoramento térmico . . . . .	9
2.3 Previsão de superaquecimentos . . . . .	10
2.3.1 Mercury . . . . .	10
2.3.2 ThermoCast . . . . .	11
2.3.3 Weatherman . . . . .	12
2.4 Detecção de anomalias . . . . .	12
2.4.1 Baseados em SAX . . . . .	13
2.4.2 Baseados em ACP . . . . .	14
<b>3 Rede sem-fio de monitoramento térmico</b>	<b>16</b>
3.1 Projeto . . . . .	16

---

3.2	Implementação . . . . .	18
3.3	Rede de coleta . . . . .	19
3.4	Economia de energia . . . . .	20
3.5	Programação sem-fio . . . . .	22
3.6	Ambiente de testes . . . . .	23
<b>4</b>	<b>Séries temporais</b>	<b>26</b>
4.1	Séries temporais univariadas . . . . .	27
4.2	Séries temporais multivariadas . . . . .	27
4.2.1	ACP - Análise de Componentes Principais . . . . .	28
4.2.2	SPIRIT . . . . .	31
4.2.2.1	Definição . . . . .	31
4.2.2.2	Variação dinâmica do número de variáveis ocultas . . . . .	33
4.2.2.3	Centralização das séries . . . . .	34
<b>5</b>	<b>Detecção de Anomalias</b>	<b>35</b>
5.1	Detecção de anomalias com o SPIRIT . . . . .	37
<b>6</b>	<b>Previsão de superaquecimentos</b>	<b>39</b>
6.1	Naïve . . . . .	39
6.2	Suavização Exponencial . . . . .	40
6.2.1	Média Exponencial . . . . .	40
6.2.2	Método Linear de Holt . . . . .	41
6.3	Autoregressão . . . . .	42
6.4	Perceptrons . . . . .	43
6.4.1	Perceptron Linear . . . . .	43
6.4.2	Perceptron Linear com Rolamento . . . . .	43
6.5	Erros . . . . .	44

---

6.5.1	Ajuste de parâmetros . . . . .	45
6.6	Proposta . . . . .	46
<b>7</b>	<b>Experimentos</b>	<b>49</b>
7.1	Detecção de anomalias . . . . .	49
7.1.1	Ambiente de testes . . . . .	49
7.1.2	Experimentos . . . . .	51
7.1.2.1	Experimento I . . . . .	51
7.1.2.2	Experimento II . . . . .	52
7.1.3	Conclusão . . . . .	54
7.2	Previsão de superaquecimentos . . . . .	56
7.2.1	Ambiente de testes . . . . .	57
7.2.2	Ajuste de parâmetros dos modelos . . . . .	58
7.2.3	Acurácia de previsão dos modelos . . . . .	60
7.2.4	Análise do tempo de antecedência . . . . .	62
7.2.5	Conclusão . . . . .	64
<b>8</b>	<b>Conclusão</b>	<b>67</b>
8.1	Trabalhos futuros . . . . .	68
	<b>Referências</b>	<b>69</b>

# Lista de Figuras

1.1	Esquema de corredores frios e quentes . . . . .	3
2.1	Modelagem de um CPD usando a ferramenta TileFlow . . . . .	8
2.2	Fluxo de ar em um CPD [31] . . . . .	8
2.3	Uma série temporal (linha cinza fina) é discretizada em intervalos fixos. Em seguida, a série é quantizada em três níveis correspondentes ao alfabeto $\{a, b, c\}$ . A palavra SAX resultante é <b>cbccbaab</b> [19]. . . . .	13
3.1	Hardware utilizado na rede de sensores . . . . .	19
3.2	Cluster Oscar – frente . . . . .	24
3.3	Cluster Oscar – trás . . . . .	24
3.4	Cluster Oscar – lateral . . . . .	25
4.1	Gráfico <i>line chart</i> representando a dívida externa bruta brasileira . . . . .	27
5.1	Eletrocardiograma com a presença de uma arritmia . . . . .	36
6.1	Esquema de previsão . . . . .	47
7.1	Experimento I – <i>trace</i> . . . . .	53
7.2	Experimento I – anomalias esperadas e detectadas . . . . .	53
7.3	Experimento II – <i>trace</i> . . . . .	55
7.4	Experimento II – anomalias e eventos . . . . .	55
7.5	<i>Trace</i> de temperatura durante a falta de energia . . . . .	57
7.6	Erros em previsões – AdaptableAR . . . . .	59
7.7	Erros em previsões – AdaptableHolt . . . . .	60
7.8	Erros em previsões – LinearPerceptron . . . . .	61
7.9	Erros em previsões – RollingPerceptron . . . . .	62

---

7.10 Erros de temperatura em várias janelas de previsão . . . . .	63
7.11 Tempo de antecedência e alarmes falsos para vários previsores . . . . .	65

# Lista de Tabelas

4.1	Notação . . . . .	26
7.1	Posicionamento dos <i>motes</i> . . . . .	50
7.2	Intervalos dos eventos térmicos . . . . .	51
7.3	Anomalias do Experimento I . . . . .	52
7.4	Anomalias do Experimento II . . . . .	54
7.5	Localização dos <i>motes</i> no ambiente de testes . . . . .	58
7.6	Sumário dos resultados da previsão térmica . . . . .	65

# Capítulo 1

## Introdução

Nos últimos anos a demanda por aplicações como redes sociais, vídeo sob demanda e motores de busca vêm crescendo ao redor do mundo. A era da computação móvel trouxe uma mudança de paradigma, onde *smartphones* agem como clientes magros (*thin clients*) para aplicações que vivem na nuvem. Essa tendência continua com a emergência dos dispositivos vestíveis (*wearable devices*), como o Google Glass [15] e Pebble Smartwatch [41]. A fim de acompanhar essa demanda, a infra-estrutura de suporte, composta de *datacenters* precisa ser expandida.

Um *datacenter* ou CPD (Centro de Processamento de Dados) é o local que hospeda a infraestrutura de TI para processar e armazenar conteúdo de uma dada organização. CPDs têm sido um foco de pesquisa na última década devido a grande quantidade de energia necessária para mantê-los. Essa energia muitas vezes é produzida através da queima de combustíveis fósseis (i.e. carvão, gás e petróleo) emissores de gases do efeito estufa. Estima-se que essa indústria será responsável por 4% das emissões globais de CO<sub>2</sub> até 2020 [9].

Uma estimativa realizada no ano de 2010 indica que *datacenters* são responsáveis pelo consumo de 1,3% de toda energia elétrica do mundo [21]. Grande parte dessa energia é consumida pelo sistema de arrefecimento, uma vez que 50% da energia gasta nesses centros destina-se ao resfriamento dos equipamentos e distribuição de energia<sup>1</sup> [4]. Em um pequeno CPD com área de 280 m<sup>2</sup>, o custo anual de resfriamento pode chegar a dezenas de milhões de dólares [36].

O sistema de resfriamento de um *datacenter* se faz necessário pois uma característica

---

<sup>1</sup>O sistema de distribuição de energia geralmente envolve alimentação reserva, regulação de voltagem e conversão entre corrente alternada e contínua.

dos equipamentos de TI – principalmente servidores – é a grande dissipação de calor e a sensibilidade desses equipamentos à temperatura. A temperatura elevada pode diminuir o tempo de vida de um componente ou até levá-lo à falha. Além dos prejuízos causados pela perda do hardware, a interrupção de um serviço pode custar bem mais caro. Em outubro de 2013, o famoso *website* de compras Amazon.com saiu do ar por cerca de 30 minutos. De acordo com uma estimativa calculada pela revista eletrônica Forbes [13], o prejuízo foi de quase 2 milhões de dólares. O *downtime*, de forma geral, também gera danos à reputação da empresa e a experiência de seus usuários.

Segundo Brown et al. [4], mais de 70% dos operadores de CPDs identificaram o sistema de arrefecimento como a principal causa de problemas no gerenciamento da instalação. A expectativa de falhas acarreta em um aumento da capacidade de resfriamento dos *datacenters* e gasto energético resultante. Por outro lado, as empresas são pressionadas a economizar energia para diminuir o impacto no meio-ambiente e manterem-se competitivas no mercado, reduzindo os custos de operação. É notório que a causa principal da falta de eficiência nesses centros é a falta de visibilidade do seu perfil térmico [26]. Além disso, ao elevar a temperatura do CPD é necessário assegurar que flutuações de temperatura não ocasionarão superaquecimentos. É nesse contexto que surge a necessidade de um sistema de monitoramento térmico para *datacenters*.

## 1.1 Ambiente de um CPD

A infraestrutura de um CPD é composta de vários sistemas como distribuição de energia, anti-incêndio, rede de comunicação e energia de reserva através de um UPS – *Uninterruptible Power Supply*. Servidores são organizados verticalmente em bastidores (*racks*) que contêm geralmente 42 U (uma altura de aproximadamente 2 metros). O *rack unit* (U) é uma unidade padronizada de altura para equipamentos que são montados em bastidores (servidores podem ter 1 U ou mais). Um CPD de larga escala pode ter mais de mil bastidores<sup>2</sup>.

Unidades de refrigeração chamadas CRAC (*Computer Room Air Conditioning Unit*) são instaladas na periferia da sala de servidores. Esses aparelhos têm a função de coletar ar quente, próximo ao teto da sala, resfriá-lo, e injetar ar refrigerado no ambiente. O piso é elevado e possui uma área que permite a passagem de ar. Sendo assim, o ar

---

<sup>2</sup>Um exemplo disso é o *datacenter* de Tamboré, São Paulo, da empresa ALOG. A unidade possui 1.600 m<sup>2</sup> de área de piso elevado e capacidade para 1.260 bastidores. Além disso, possui capacidade de refrigeração de aproximadamente 24 milhões de BTUs [1].

frio proveniente dos CRACs passa por debaixo do piso e chega até à parte frontal dos servidores através de pequenos furos no piso.

Os bastidores são organizados em filas, que por sua vez são organizadas em corredores quentes e frios. Esse leiaute visa diminuir a mistura de ar frio fornecido pelas unidades CRAC com o ar quente expelido pelas ventoinhas dos servidores. Esse esquema funciona arranjando as filas de servidores de modo espelhado, como mostra a Figura 1.1<sup>3</sup>.

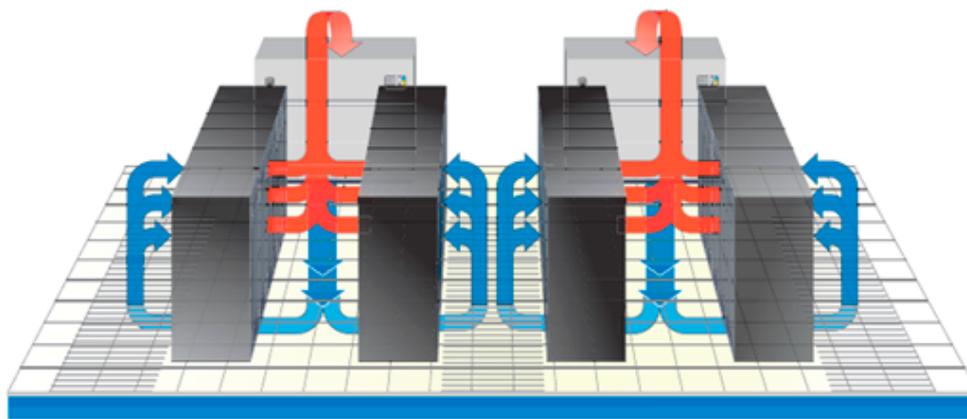


Figura 1.1: Esquema de corredores frios e quentes

## 1.2 Problemas térmicos comuns em CPDs

Em muitos *datacenters*, o sistema de arrefecimento não é otimizado. Os arquitetos da instalação projetam a potência do equipamento de resfriamento de acordo com informações providas pelos fabricantes (*nameplate*). No entanto, os valores indicados pelos manuais desses equipamentos correspondem a situações extremas (e.g. utilização máxima do processador) ou são superestimados [11]. Além disso, os arquitetos instalam uma capacidade adicional para comportar a expansão futura do CPD. Essa super-provisão de capacidade de arrefecimento se traduz em menor eficiência energética e, portanto, um maior custo de operação e impacto ambiental.

A ineficiência térmica também pode ter como fonte o leiaute da sala e dos bastidores. Uma técnica comumente utilizada é a de corredores frios e quentes, mencionada anteriormente. No entanto, mesmo com essa organização, é possível observar recirculação de ar caso não haja um isolamento total entre os corredores. A recirculação é prejudicial pois mistura ar quente dos exaustores com ar frio, diminuindo a capacidade de refrigeração do CRAC. Por esse motivo, utiliza-se muitas vezes painéis de isolamento no teto das filas de

<sup>3</sup>Figura retirada de <http://www.canadiancolo.com/cooling.html>.

servidores.

O **desbalanceamento térmico** é um problema causado por qualquer agente que crie uma distribuição não uniforme de temperatura no CPD. Isso pode ocorrer naturalmente dado que os servidores podem ter tipos de hardware diferentes e com níveis de dissipação de calor distintos. Além disso, uma distribuição irregular de carga entre bastidores ou um fluxo de ar menos intenso em algumas regiões também pode causar desbalanceamento térmico. A causa deste último pode ser o mal funcionamento de um CRAC ou uma obstrução na saída de ar. De qualquer forma, esse tipo de problema acarreta em um uso ineficiente do sistema de refrigeração ao impor que a potência das unidades CRAC seja aumentada para resfriar um subconjunto de máquinas mais quentes que as demais.

Um caso mais extremo de desbalanceamento térmico ocorre quando uma pequena área se torna mais quente que o restante da sala, formando a chamada **ilha-de-calor** (ou *hotspot*). Tal fenômeno pode ocorrer por conta de um efeito local, como a falha de uma ventoinha de um servidor, o bloqueio de um exaustor (e.g. por conta de poeira acumulada) ou a ocupação da CPU por conta de *bugs* de software. Além disso, a tendência de consolidar carga de trabalho em poucas máquinas de forma a evitar a subutilização de servidores também pode ser uma origem de ilhas-de-calor.

Outra preocupação em relação ao sistema de arrefecimento são os chamados **superaquecimentos** (utiliza-se esse termo para *redlinings*). Eles ocorrem quando a temperatura de um componente ultrapassa um limiar considerado seguro. Tal evento pode diminuir consideravelmente o tempo de vida do equipamento. Um estudo aponta que a cada 10 °C acima de 21 °C, o tempo de vida de disco rígidos diminui pela metade [35]. Todavia, a determinação do nível de *redline* é controverso. Alguns artigos mais recentes apontam uma temperatura alvo de 35 °C [10], 10 °C maior do que o valor que era frequentemente aceito [44]. A posição defendida nesse caso é que a diminuição do tempo de vida do equipamento compensaria a economia de energia resultante.

Casos extremos de *redlining* passam a ser chamados de *emergências térmicas*. Essa situação ocorre quando a temperatura ultrapassa o limiar físico de um componente, levando à falha (temporária ou não) do mesmo. No caso de processadores, a maioria conta com mecanismos de controle que procuram evitar essa situação através da diminuição da frequência do processador ou o seu desligamento imediato [38]. No entanto, a partir desse ponto o dispositivo não pode mais ser usado, o que gera prejuízo para a empresa ou organização responsável pelo CPD.

## 1.3 Proposta

Este trabalho propõe uma solução responsiva para o monitoramento térmico de CPDs, pois reage rapidamente à mudanças no perfil térmico. Isso possibilita que operadores do *datacenter* ou outros sistemas atuem prontamente. Esse projeto foi validado através de experimentos em um ambiente físico de um *datacenter*. De forma resumida, pode-se dividi-lo em três módulos:

**Rede de monitoramento térmico.** É fundamental obter o mapa térmico de um CPD a fim de identificar desbalanceamentos térmicos. Para tanto, emprega-se uma rede de sensores autônoma e independente da infraestrutura do CPD. Tal rede necessita de pouca manutenção uma vez que apresenta um grande tempo de vida (mais de 40 dias).

**Deteção de anomalias térmicas.** O ambiente do *datacenter* é extremamente errático do ponto de vista termodinâmico. É importante alertar os operadores do CPD para situações inesperadas que podem estar relacionadas a fenômenos como ilhas de calor e ineficiências térmicas no momento em que elas surgem. Além disso, é importante indicar a região afetada para uma rápida análise da causa do problema. Para essa tarefa empregou-se um detector de anomalias que foi capaz de identificar satisfatoriamente mudanças bruscas no perfil térmico durante os experimentos realizados.

**Previsão de superaquecimentos.** O problema da super-provisão muitas vezes é uma consequência de uma política agressiva para evitar superaquecimentos. Isso ocorre pois o mapa térmico da instalação não é levantado ou então os operadores não dispõem de ferramentas confiáveis para alertá-los caso um superaquecimento seja iminente. Uma contribuição deste trabalho é uma ferramenta que permite prever tais situações com antecedência suficiente para que operadores possam evitá-las.

## 1.4 Divisão de capítulos

O restante desta dissertação está dividida da seguinte forma: o Capítulo 2 traz um levantamento da literatura relevante sobre o tema e contextualiza este trabalho. Em seguida, no Capítulo 3, discutem-se as características que a rede de monitoramento deve atender e apresentam-se uma série de tecnologias usadas para implementá-las. O Capítulo 4 fundamenta a base matemática utilizada pelas aplicações de deteção de anomalias e previsão

---

térmica e introduz uma solução para reduzir a alta dimensionalidade das séries temporais (variáveis monitoradas). Em seguida, o Capítulo 5 apresenta o método eleito para realizar a detecção de anomalias térmicas e a sua motivação. O Capítulo 6 tem a função de mostrar uma série de abordagens para a previsão de superaquecimentos. A eleição de qual abordagem é mais indicada foi deixada para o capítulo seguinte. Por sua vez, o Capítulo 7 consiste de uma análise sobre a validação dos métodos propostos e a metodologia utilizada. Por fim, o Capítulo 8 conclui este trabalho e discute alguns caminhos que podem ser tomados para estendê-lo.

# Capítulo 2

## Trabalhos relacionados

Neste capítulo serão apresentadas as pesquisas mais relevantes para o tema desta dissertação. O capítulo foi dividido em quatro partes. A Seção 2.1 discute os trabalhos sobre DFC, uma área que impulsionou os primeiros trabalhos sobre monitoramento térmico, mas que no entanto não é executada em tempo real. Em seguida, mostra-se uma abordagem para monitoramento térmico através de uma rede de sensores, que serviu de inspiração para este trabalho, na Seção 2.2. A Seção 2.3 traz pesquisas que lidam com a previsão do perfil térmico de CPDs. Por fim, a Seção 2.4 discute abordagens para detecção de anomalias, com base nas técnicas SAX e ACP.

### 2.1 Dinâmica dos Fluidos Computacional

Dinâmica dos Fluidos Computacional (DFC) é um conjunto de métodos capazes de obter soluções aproximadas de problemas que envolvem transferência de calor e dinâmica de fluidos (fluxo de ar) através de métodos numéricos [49].

No que se refere ao escopo deste trabalho, deseja-se analisar ferramentas que, através da DFC, possam fornecer informações sobre o estado térmico de um *datacenter*, a interação do ar com os equipamentos e avaliar o sistema de refrigeração da instalação. Tais simuladores também são utilizados para avaliar cenários, como o impacto de alguma política de economia de energia. Dentro desse contexto, algumas ferramentas comerciais são: TileFlow [18], CoolSim [2] e FloVent [33].

A Figura 2.1 mostra resultados de alguns dos relatórios obtidos utilizando a ferramenta TileFlow durante um período de demonstração. Nela tem-se um *datacenter* composto por duas fileiras de 5 bastidores e uma unidade de ar condicionado cuja temperatura de saída

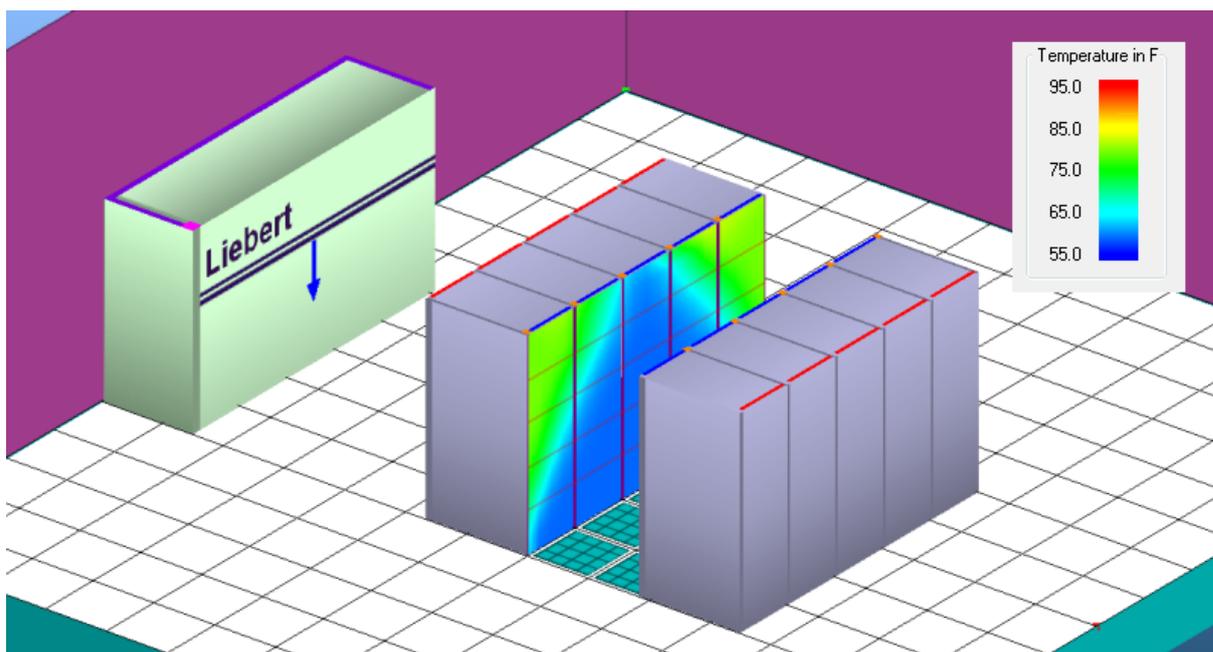


Figura 2.1: Modelagem de um CPD usando a ferramenta TileFlow

está ajustada para 15 °C. A estrutura desse ambiente é a de corredores quentes e frios, onde o corredor frio se encontra entre as duas fileiras. O resultado da simulação mostra a temperatura de cada um dos bastidores através de um espectro de cores referentes à legenda da figura.

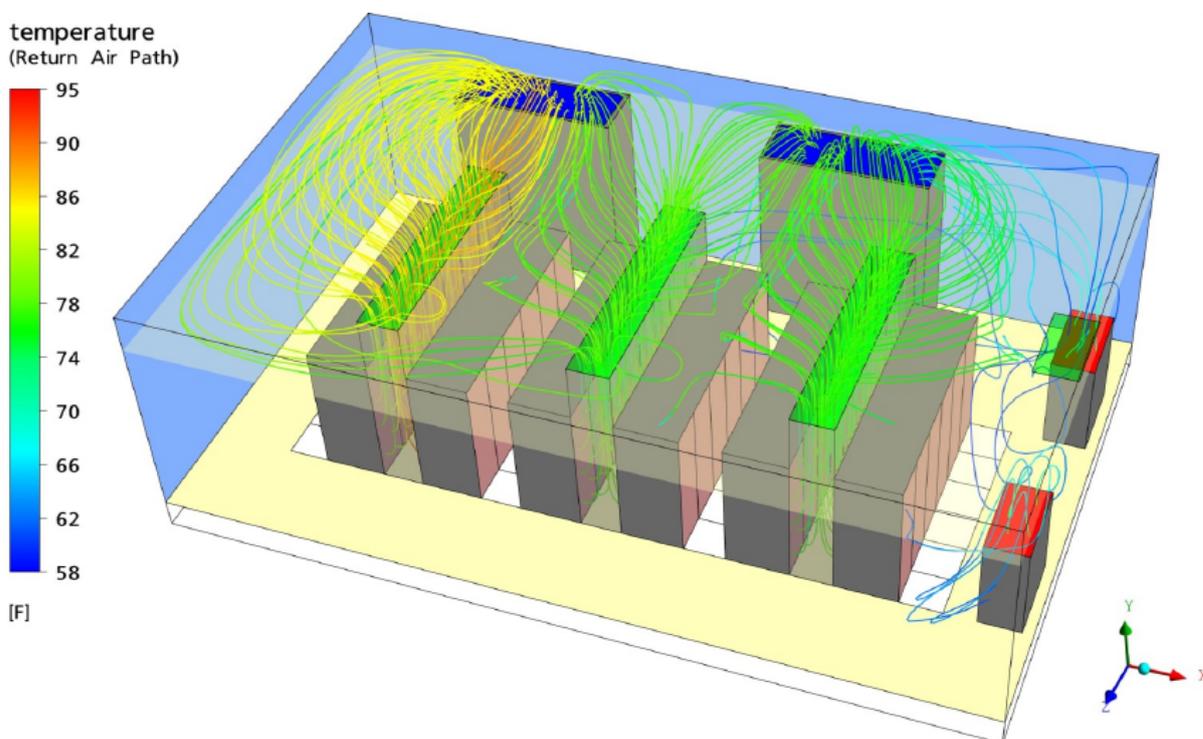


Figura 2.2: Fluxo de ar em um CPD [31]

Os métodos de DFC são capazes de lidar com *datacenters* estruturalmente mais complexos. A Figura 2.2 mostra o comportamento do fluxo de ar quando existe um teto falso na planta do ambiente em questão (representado como um plano translúcido). O resultado de tal recurso diminui o efeito da recirculação de ar dentro do ambiente.

Apesar de possuírem vários recursos, as ferramentas de DFC não são capazes de simular o ambiente de um *datacenter* em tempo real, pois são voltadas para o tempo de projeto da instalação. Os primeiros trabalhos da área utilizavam ferramenta de DFC para desenvolver políticas de alocação de carga computacional e testar cenários onde há falha de unidades CRAC [36, 44].

Softwares DFC são caros e necessitam de grande poder computacional. Uma licença comercial de 12 meses do software TileFlow custava aproximadamente US\$ 17,000, segundo uma cotação realizada em dezembro de 2011. Além disso, outros produtos, como o CoolSim, utilizam até servidores na *nuvem* para realizar as simulações.

## 2.2 Monitoramento térmico

Diversas abordagens da literatura utilizam redes de sensores para coletar dados e realizar monitoramento térmico [25, 26, 50]. A motivação dessas abordagens varia entre prevenção de superaquecimento dos componentes do CPD e economia de energia através de consolidação de servidores e balanceamento de carga.

A RACNet [26] é uma rede de monitoramento térmico para *datacenters* desenvolvida pela Microsoft Research. A rede física consiste de dispositivos desenvolvidos especificamente para o projeto, denominados Genomotes. Existem dois tipos de Genomote: sem-fio e sensoramento. Os dispositivos se comunicam através de uma topologia *daisy chain*, onde vários nós de sensoramento se conectam a um nó sem-fio que, por sua vez, provê a comunicação entre outras estações do mesmo tipo.

No mesmo trabalho, os autores apresentam o protocolo WRAP – *Wireless Reliable Acquisition Protocol* que roda sob os nós sem-fio. Esse protocolo tem a missão de combater problemas comuns em redes desse tipo: a alta taxa de colisões (devida à densidade da rede) e grande atenuação do sinal (causada pelo ambiente do CPD). Para tanto, são construídas diversas árvores de coleta que são balanceadas entre múltiplos canais do espectro de radiofrequência. Tais árvores são balanceadas de forma a não sobrecarregar nenhum canal e *notes* são encorajados a mudar de canal caso detectem uma alta taxa de perdas. A coleta em si visa evitar colisões de mensagens através de um protocolo de passagem de

fichas (*tokens*). Só existe uma ficha na rede por vez e cada nó só pode transmitir quando está de posse da mesma. Ao receber a confirmação do envio, um nó repassa a ficha até que o ciclo de coleta termine.

Nesta dissertação, utiliza-se um hardware similar aos Genomotes sem-fio, que também são compatíveis com o TinyOS (ver Seção 3.2 para mais detalhes). A alimentação dos Genomotes é realizada via tomada ou porta USB; já os motes utilizados aqui são inteiramente autônomos já que possuem alimentação via bateria. O protocolo utilizado para comunicação entre nós não é o WRAP, devido a sua grande complexidade e ausência de suporte na comunidade *open-source* do TinyOS. Além disso, o WRAP foi desenvolvido para uma rede que possui alta taxa de transmissão de dados, o que não é o caso da rede deste trabalho, que utiliza compressão de amostras.

## 2.3 Previsão de superaquecimentos

Nesta seção serão apresentados os principais trabalhos que influenciaram na escolha da abordagem de previsão térmica deste trabalho.

### 2.3.1 Mercury

O Mercury é um modelo simples para previsão de temperatura de componentes de um servidor [16] que utiliza dígrafos para representar a influência do fluxo de ar entre os componentes do servidor. Para gerar este modelo, é necessário especificar atributos difíceis de obter, como o calor específico e massa de dispositivos (e.g., processador e disco). De posse dessas informações, o Mercury utiliza equações termodinâmicas para modelar a temperatura em função da utilização dos dispositivos. Além disso, os autores utilizaram um software DFC para estimar os pesos associados às arestas do dígrafo. Apesar do *profiling* trabalhoso, os autores conseguiram estimar a temperatura com precisão de menos de 1 °C baseado apenas na utilização dos componentes.

O Mercury foca na modelagem térmica e fluxo de ar dos componentes internos de um servidor e não do ambiente como um todo. Outra pesquisa [26] aponta que a temperatura interna do servidor não é útil para analisar as condições térmicas do ambiente e, sendo assim, uma abordagem como o Mercury não pode ser utilizada para prever superaquecimentos.

### 2.3.2 ThermoCast

Um trabalho mais recente propõe uma abordagem de previsão baseada não somente nos dados (*data-driven*) mas também em um modelo termodinâmico [25]. O modelo, chamado ThermoCast, é baseado em equações básicas da termodinâmica que são válidas em pequenas zonas ao redor de um bastidor. O aprendizado dos parâmetros é feito de maneira a minimizar o erros entre previsões e as amostras de temperatura. Para contabilizar efeitos não-estacionários como desligamentos de servidores, é utilizada uma média exponencial sobre as medidas de erro. A fim de estimar o fluxo de ar do piso, pois o mesmo é permitido ser variável, empregou-se um modelo autorregressivo.

A avaliação do ThermoCast foi realizada utilizando um modelo AR(2), puramente baseado em dados, como *baseline*. Através dos experimentos realizados constatou-se uma melhoria no número de alarmes térmicos previstos e principalmente uma antecedência maior sobre a previsão desses alarmes. Mais precisamente, o ThermoCast é capaz de prever, em média, alarmes com 4,2 minutos de antecedência enquanto o modelo AR é de apenas 2,3 minutos. Argumenta-se que o ganho de cerca de 2 minutos é relevante, já que a atuação no sistema (e.g. suspender uma máquina virtual) pode levar alguns minutos para ter efeito.

Tanto o ThermoCast quanto o modelo AR(2) são treinados uma vez em uma janela de 90 minutos e em seguida são avaliados. Essa abordagem desconsidera mudanças no perfil térmico que invalidem os resultados obtidos durante a fase de treinamento. Mais importante, os autores pressupõem que situações de superaquecimento estão disponíveis para realizar o treinamento e que superaquecimentos subsequentes terão um perfil térmico similar, o que são situações não-realistas. Ao invés disso, esta dissertação considera técnicas que ajustam automaticamente seus parâmetros e utilizam aprendizado em tempo de execução.

Os dados utilizados no trabalho são coletados de um CPD da Carnegie Mellon University durante a sua operação normal. No entanto, por conta da falta de superaquecimentos no conjunto de dados, os autores impuseram um limiar de 16 °C acima do qual o sistema seria declarado como superaquecido. Não é claro que essa situação pode ser extrapolada para um superaquecimento real. Já o conjunto de dados para previsão térmica deste trabalho é fruto de um superaquecimento real, onde a temperatura de alguns sensores chega a mais de 50 °C devido a falha no sistema de arrefecimento.

O modelo AR(2) apresentado no artigo original do ThermoCast foi incorporado na

análise deste trabalho uma vez que foi comprovada a sua eficácia em obter um tempo de antecedência de mais de 2 minutos em uma janela de previsão de 5 minutos. Porém, o ajuste de parâmetros do modelo é feito de forma iterativa (para mais detalhes, veja a Seção 6.3).

### 2.3.3 Weatherman

Uma das primeiras técnicas de previsão do perfil térmico de *datacenters* foi chamada de Weatherman [35]. Apontou-se a não-linearidade do sistema termodinâmico de um *datacenter* como motivação para utilizar uma técnica de Aprendizagem de Máquina, como Redes Neurais Artificiais [48]. O objetivo dessa rede é “aprender” o perfil térmico resultante de uma série de distribuições de carga. Estes são chamados de dados de treinamento e através deles a rede neural é capaz de prever perfis térmicos para distribuições de cargas desconhecidas.

O Weatherman conseguiu um erro sobre a temperatura prevista de menos de 1 °C em 92% dos casos. Uma das desvantagens da proposta é a quantidade de dados necessários para fazer o treinamento. Os autores obtiveram seus dados de treinamento a partir de um software DFC, que por sua vez pode ser manipulado para emular diversos cenários distintos favorecendo o treinamento da rede. Porém, situações de *redlining* podem estar ausentes do conjunto de dados de treinamento em uma aplicação prática.

Outra questão é sobre a escalabilidade da rede neural. Com o crescimento do tamanho do *datacenter*, mais neurônios são necessários na rede, fazendo com que o custo de treinamento e de computação de previsões também cresça. Uma abordagem adotada pelo trabalho para mitigar esse efeito foi agrupar servidores próximos em um único bloco e realizar a previsão sobre eles ao invés de servidores individuais. No entanto, a relação entre os tamanhos dos blocos e a acurácia das previsões não foi investigada.

## 2.4 Detecção de anomalias

A maioria das abordagens para detecção de anomalias na literatura não visa séries temporais [7]. Entre aquelas que são voltadas para análise temporal, muitas não são capazes de lidar com séries multivariadas [19, 47]. Nesta seção serão discutidos alguns métodos consolidados na literatura para a tarefa de detecção de anomalias. Uma discussão mais abrangente sobre o tema de detecção de anomalias será delegada ao Capítulo 5.

### 2.4.1 Baseados em SAX

O padrão *de facto* para mineração de séries temporais é o SAX – *Symbolic Aggregate approXimation* [27]. Trata-se de uma técnica para quantização e compressão de séries temporais em uma representação intuitiva que evita a distorção da série temporal. Isso é feito a fim de diminuir o esforço computacional requerido para realizar o processo de mineração (e.g., classificação e clusterização) sobre os dados originais. No entanto, resultados derivados a partir de uma representação SAX podem ser generalizados para a série original em muitos casos.

O processo funciona através de uma definição de um alfabeto, como  $\{a, b, c\}$ , cujos elementos correspondem aos níveis de quantização. Além disso, define-se uma janela sobre a qual as medidas da série temporal serão substituídas por uma letra do alfabeto. Uma sequência de letras forma uma palavra, que representa um trecho da série temporal, como indicado na Figura 2.3.

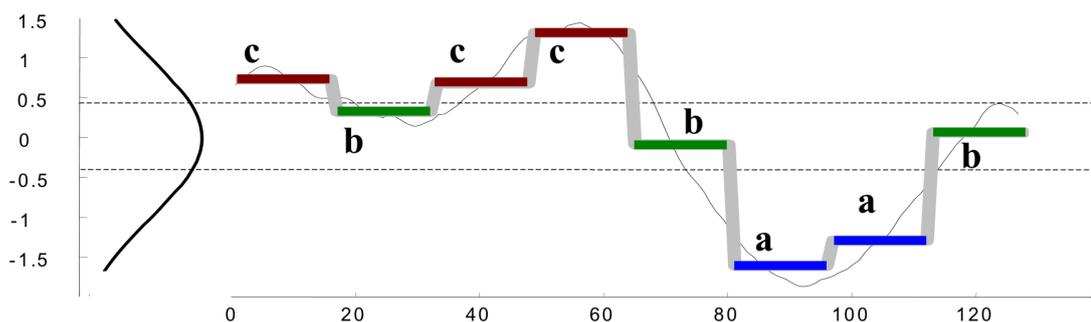


Figura 2.3: Uma série temporal (linha cinza fina) é discretizada em intervalos fixos. Em seguida, a série é quantizada em três níveis correspondentes ao alfabeto  $\{a, b, c\}$ . A palavra SAX resultante é **cbccbaab** [19].

A abordagem auto-intitulada Escalation [50] é um sistema distribuído de detecção de anomalias (no artigo original, os autores adotaram o termo “eventos complexos”) em tempo real. Cada unidade de sensoriamento amostra periodicamente dados do ambiente e converte a série temporal na representação SAX. Além disso, o sensor automaticamente decide alterar a taxa de amostragem baseado na dispersão que observa no ambiente.

O Escalation requer um intervalo inicial de treinamento, onde irá criar um *baseline* do que é considerado o comportamento normal do sistema. Isso é feito medindo a distância máxima entre palavras SAX adjacentes. Após a fase de treinamento, distâncias maiores que o valor *baseline* são reportadas como anomalias.

Outra abordagem, intitulada HOT SAX – *Heuristically Ordered Time series using SAX* [19], tem uma outra definição para anomalia: a subsequência da série temporal

menos frequente. Assim como outras abordagens SAX, utiliza poucos parâmetros, sendo o único parâmetro relevante para ser especificado o tamanho da subsequência a ser avaliada. Enquanto esse parâmetro é fácil de ser especificado para aplicações como detecção de anomalias em um eletrocardiograma, outras aplicações necessitarão de um ajuste empírico. A proposta apresenta uma heurística que evita uma busca exaustiva entre todas as combinações de pares de subsequências possíveis, resultando em um tempo de pesquisa 3 mil vezes mais rápido. No entanto, é necessário aplicar o algoritmo na série temporal completa, um cenário incompatível com séries medidas em tempo real.

O *Times series Bitmap* (TB) [22, 47] é uma abordagem que captura a frequência de subsequências SAX em um artefato denominado *bitmap*. O mecanismo proposto funciona através *bitmaps* que são criados sobre duas janelas deslizantes: *anterior* e *posterior*. A primeira, geralmente mais longa que a segunda, tem a função de capturar o comportamento considerado normal da série temporal. Já a janela posterior é usada para representar o estado atual do sistema e detectar anomalias. Uma desvantagem da proposta é que as anomalias são detectadas com um atraso igual ao tamanho da janela posterior. No entanto a abordagem é considerada intuitiva e os *bitmaps* podem ser utilizados para distinguir visualmente diferenças entre séries temporais.

Tanto HOT SAX quanto TB são abordagens que trabalham com uma série temporal de apenas uma variável. Em um cenário de múltiplas séries temporais seria necessário rodar o algoritmo em cada série separadamente, resultando em um custo computacional proibitivo. Enquanto é possível utilizar uma abordagem distribuída, como é o caso do Escalation, onde cada sensor é responsável por detectar anomalias, essa abordagem não é adequada para o cenário deste trabalho. A análise individual é passível de *outliers* locais em uma rede com muitos sensores. Mesmo na ausência de *outliers*, o operador precisa interpretar a causa da anomalia sabendo apenas que ocorreu uma anomalia em um dado sensor. Para a aplicação de monitoramento térmico, deseja-se detectar anomalias levando em conta a informação do perfil térmico global e todos os sensores presentes.

## 2.4.2 Baseados em ACP

O método ACP (Análise de Componentes Principais) é utilizado para redução dimensional. No caso de múltiplas séries temporais, o ACP permite trabalhar com um conjunto menor de séries que representam o comportamento geral do processo. Mais detalhes sobre o ACP estão na Seção 4.2.1.

O InteMon [17] é um sistema de monitoramento de fácil configuração para *datacenters*.

O sistema proposto monitora uma série de variáveis por servidor, como utilização de CPU e disco e também tráfego de rede. Além disso, integra-se às unidades de refrigeração para monitorar umidade e temperatura de entrada e saída. Todas essas variáveis servem de entrada para um algoritmo apresentado em um outro trabalho, chamado SPIRIT [40] que é um estimador em tempo real do ACP (para mais detalhes, veja a Seção 4.2.2). Além de detectar anomalias, o sistema é capaz de comprimir os dados monitorados, reduzindo a capacidade necessária para armazenamento, e preencher dados faltantes (por conta de erros de transmissão) através de dados históricos e de outras séries.

Uma outra abordagem baseada em ACP é o Holmes [46]. Trata-se de um sistema criado para monitorar diversas séries temporais com a missão de detectar falhas e problemas que possam afetar a experiência dos usuários ou a segurança da aplicação do *datacenter*. Enquanto o InteMon utiliza *queries* periódicas aos servidores para obter medidas, o Holmes adota uma coleta de dados orientada à eventos (*publish-subscribe*). O sistema também foi testado em uma aplicação real, o sistema de votação Big Brother Brasil, cuja infraestrutura possuía cerca de 15 mil itens sendo monitorados em tempo real.

Ambas abordagens requerem uma integração e adaptação com a aplicação do *datacenter*. Além disso, utiliza-se a própria infra-estrutura da instalação para realizar o sensoriamento. Neste trabalho, busca-se uma abordagem que seja simultaneamente autônoma e agnóstica em relação a aplicação que roda no CPD.

# Capítulo 3

## Rede sem-fio de monitoramento térmico

Neste capítulo introduz-se a rede de monitoramento térmico implementada neste trabalho. Primeiramente apresenta-se o projeto e os requisitos fundamentais que a rede precisa atender na Seção 3.1. Na Seção 3.2 é dado um panorama sobre as tecnologias e dispositivos de hardware que compõem a rede. Em seguida, detalha-se o funcionamento da rede de coleta de amostras de temperatura na Seção 3.3. Outro aspecto importante é a autonomia da rede e por conta disso estudou-se algumas abordagens de economia de energia na Seção 3.4. Dada a grande quantidade de nós que a rede pode conter, a manutenção do software embarcado pode ser trabalhosa e por essa razão utilizou-se um protocolo de programação sem-fio que é explicado na Seção 3.5. Por fim, descreve-se o local onde foram realizados os testes da rede de sensores na Seção 3.6.

É importante notar que a rede descrita aqui foi idealizada e projetada de forma conjunta com outro trabalho [5, 6]. Sendo assim, o leitor que desejar se aprofundar em algum aspecto descrito neste capítulo pode consultar os trabalhos citados para mais detalhes.

### 3.1 Projeto

Sensores de temperatura são encontrados em vários locais de um servidor moderno. Usualmente estão localizados na placa-mãe, dispositivos externos (como discos rígidos) e na CPU. No entanto, esses sensores refletem a carga de trabalho que o servidor está sujeito e não são úteis para identificar problemas térmicos globais no ambiente da instalação [26]. Um problema mais grave, é que o uso desses sensores implica em manter os servidores ligados. Sendo assim, ao desligar um servidor (por motivo de economia de energia ou manutenção), o sistema ficaria sem a informação sobre temperatura daquele local.

O escopo do monitoramento térmico deve ser toda a instalação, de forma a capturar as interações entre o sistema de arrefecimento e o equipamento de TI do local. Além disso, as condições de operação de uma sala de servidores são usualmente definidas em contrato sobre o IAT (*intake air temperature*), que corresponde a temperatura medida na entrada dos servidores [26]. Uma maneira de monitorar o IAT é posicionar sensores de forma espaçada na parte frontal dos bastidores, pois a temperatura não varia tão radicalmente a ponto que seja necessário monitorar servidores individualmente.

A rede de sensores pode usar a infraestrutura do próprio *datacenter* para encaminhar as medidas coletadas até uma base centralizadora. No entanto, tal abordagem traz alguns problemas. Como foi dito anteriormente, não se deve presumir que o sistema de energia da instalação estará disponível em todos os momentos. Além disso, a travessia dentro da rede interna de um *datacenter* pode ser impossível uma vez que certos domínios da instalação podem estar isolados (de forma lógica ou física) ou fazê-la resultaria em uma brecha de segurança.

A rede de sensores deve funcionar de forma **autônoma** em relação a infraestrutura e alimentação elétrica do *datacenter*. Enquanto é possível utilizar baterias para alimentar os sensores, ainda resta definir entre a utilização de uma rede de transporte cabeada ou sem-fio. A primeira opção implicaria em passar cabos em uma instalação que geralmente já é lotada: um processo caro e trabalhoso. Por esse motivo esta solução não é geralmente adotada na literatura. A solução adotada por este trabalho é o emprego de *motés*<sup>1</sup> sem-fio.

Uma outra preocupação se refere às grandezas que se deseja monitorar. Uma escolha óbvia é temperatura, que pode ser medida através de dispositivos chamados termístores, resistências que têm seu valor alterado de acordo com a temperatura. No entanto outros trabalhos utilizam outras medidas físicas como pressão, fluxo de ar, umidade [25, 26] e até métricas da aplicação, como carga de trabalho (dada em requisições por segundo, por exemplo) [17, 35, 46]. Alguns trabalhos necessitam de variáveis físicas pois utilizam um modelo termodinâmico para modelar o perfil térmico do *datacenter* [25, 26]. Porém o trabalho apresentado aqui utiliza somente modelos chamados dirigidos a dados (*black-box*) e portanto a temperatura é a única variável física de interesse.

Já a métrica de carga de trabalho traz informações úteis uma vez que permite correlacionar o calor dissipado pelos servidores com a demanda de usuários ou clientes. Porém, tal métrica é extremamente dependente do domínio da aplicação e requer uma comuni-

---

<sup>1</sup>O termo *mote* é utilizado para indicar um dispositivo de sensoriamento. Usa-se também o termo *nó sensor* (ou simplesmente *sensor* ou *nó*) de forma intercambiável.

cação entre os servidores e a rede de monitoramento. Neste trabalho, preferiu-se adotar uma postura totalmente autônoma e agnóstica em relação a aplicação do *datacenter*, de modo a manter o sistema genérico e útil para uma maior quantidade de casos de uso. No entanto, acredita-se que não há nenhum impedimento em enriquecer as aplicações de detecção de anomalias (Capítulo 5) e previsão térmica (Capítulo 6) com outras variáveis além de temperatura, se assim for desejado.

Outro fator, sobre a opção de não incluir variáveis como carga computacional, é a manutenção da privacidade de segredos de negócio da empresa responsável pelo *datacenter*. Suponha que a Empresa A, do segmento de *e-commerce*, deseje instalar o sistema de monitoramento descrito neste trabalho com a inclusão de uma métrica como o número de vendas por minuto. Além disso, suponha que esse serviço de monitoramento é provido pela Empresa B. Apesar dessa métrica parecer, a princípio, inofensiva, ela pode ser usada por concorrentes para analisar o perfil de vendas da Empresa A. Sendo assim, é necessário se valer de dispositivos legais, como contratos de confidencialidade, para garantir que a Empresa B não agirá de má fé. Tal problema pode ser um empecilho para a adoção de um serviço de monitoramento provido por terceiros, como no caso da Empresa B.

Usando o que foi discutido nesta seção, pode-se estabelecer uma rede de *motes* sem-fio que atende à essas especificações. A fixação dos *motes* pode ser feita em três níveis de altura (fundo, meio e topo), por bastidor, sem perda considerável de informação [3, 26, 34]. Sensores são posicionados tanto na parte frontal (entrada de ar) quanto na parte traseira (saída de ar) dos bastidores, pois deseja-se monitorar o IAT e também detectar discrepâncias na dissipação de calor entre regiões.

## 3.2 Implementação

Os *motes* utilizados neste trabalho são do modelo IRIS, fabricados pela Memsic [32]. A sua unidade de rádio implementa o padrão IEEE 802.15.4 (camada física e MAC), operando numa frequência de 2,4 GHz e é capaz de se comunicar com a velocidade de até 250 kbps. Sua fonte de alimentação são duas pilhas no formato AA. A placa de sensoriamento MDA100 é encaixada no IRIS e provê sensores de temperatura e luminosidade embutidos (outros sensores podem ser acoplados). Por fim, é possível prover alimentação e comunicação de dados entre um *mote* e um computador através da conexão USB da placa MIB520. Veja a Figura 3.1 para uma ilustração desses dispositivos.

O alcance de comunicação entre nós IRIS varia de acordo com obstáculos e outros

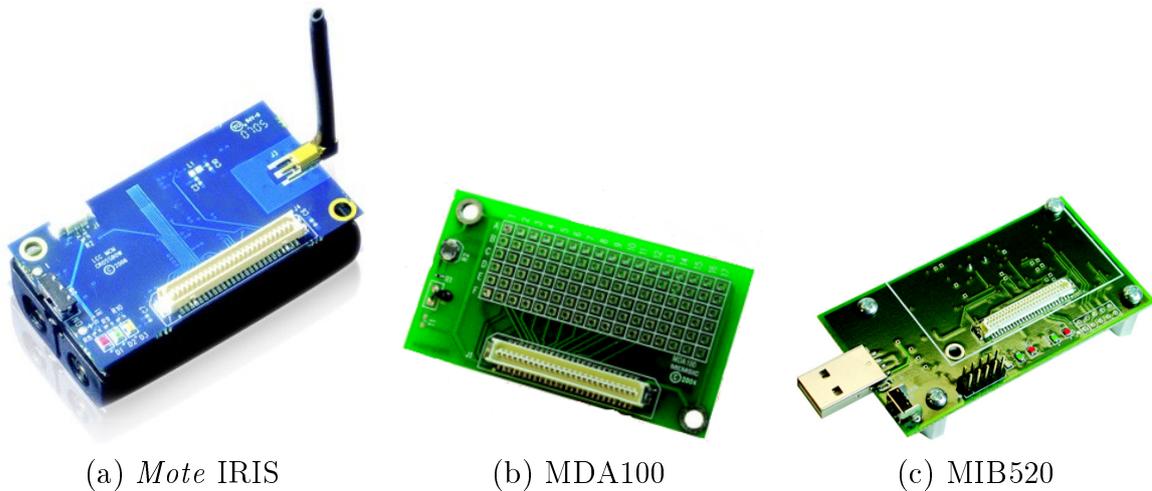


Figura 3.1: Hardware utilizado na rede de sensores

tipos de interferências. No entanto, o alcance de visão direta (i.e. sem obstáculos) em um corredor é de aproximadamente 100 m, segundo experimentos realizados.

Cada *mote* é fixado com fita adesiva dupla-face (localizada na lateral do compartimento de pilhas) sobre o chassis dos bastidores e outros locais do *datacenter*.

O IRIS é compatível com o TinyOS [24], um sistema operacional embarcado voltado para RSSFs. O sistema teve sua primeira versão publicada em 2000 e ainda é ativamente mantido pela comunidade *open-source*. O TinyOS provê uma abstração para a interação com os componentes do hardware do *mote* e permite uma programação orientada a eventos em um dialeto da linguagem C, chamada nesC. Existe uma série de bibliotecas e protocolos compatíveis com o TinyOS (utiliza-se a versão 2.1.2) que são explorados neste trabalho.

### 3.3 Rede de coleta

Cada *mote* tem um alcance limitado de comunicação. Considerando que uma instalação possui centenas de metros quadrados de área, além de obstáculos, não é factível a comunicação direta entre cada *mote* e uma base centralizadora. Por esse motivo, utiliza-se uma rede de coleta de medidas, onde a temperatura é encaminhada por múltiplos saltos até chegar a uma estação-base. O protocolo usado para essa tarefa é o CTP – *Collection Tree Protocol* [12].

Na rede de coleta existem três papéis que um *mote* pode assumir: base (*sink*), roteador e produtor. (Note que um dado *mote* pode exercer diversos papéis.) Um nó base é aquele que está conectada (através da placa MIB520) ao computador agregador de medidas. As

medidas são roteadas até uma base, que por sua vez repassa os pacotes para a porta USB do computador. Há suporte para múltiplas bases, porém este trabalho usa apenas uma.

Todos os nós da rede são roteadores, exceto bases. Um roteador tem a função de servir de ponte entre um nó e seu destino (uma base). Portanto, não é necessário nenhum componente além do *mote* IRIS. Alguns nós com apenas o papel de roteador foram posicionados de forma a criar uma ponte até a base (que fica em outra sala), neste trabalho. Já *notes* produtores têm a função de amostrar a temperatura e encaminhá-la para o próximo salto. Para realizar essa função, nós produtores contam com a placa MDA100.

Um outro tipo de sensor utilizado neste trabalho é o de carga de bateria de cada *mote*. Através desta informação é possível prever quando será necessário realizar a manutenção da rede. A voltagem atual da bateria é enviada a cada 30 min.

Uma desvantagem da comunicação sem-fio é que o ambiente físico do *datacenter* é considerado hostil para a propagação de ondas eletromagnéticas. Há uma grande concentração de material metálico, como bastidores, canos de arrefecimento e cabos que são notoriamente atenuadores de sinais de radiofrequência. Outro problema é a interferência com outros sinais de rádio que permeiam o ambiente. Tais sinais podem ter origem em outros rádios (e.g. sinal Wi-fi de um roteador próximo) ou a própria rede de sensores (auto-interferência).

Por conta desses problemas, é natural que os enlaces entre cada nó sensor tenham uma qualidade e alcance diferentes. Pensando nisso, a rede de coleta utilizada neste trabalho envia dados automaticamente pela rota que possui a menor quantidade de retransmissões esperadas. Dado que mudanças na condição dos enlaces também pode mudar com o tempo, a rede frequentemente atualiza a informação sobre essas rotas.

A rede de coleta necessita refletir, com uma certa precisão, o instante em que a medida foi coletada de forma a preservar a causalidade dos eventos térmicos que serão utilizados pela aplicação de detecção de anomalias. Sendo assim, utilizou-se um protocolo distribuído de sincronia de relógios chamado FTSP – *Flooding Time Synchronization Protocol* [30].

## 3.4 Economia de energia

Uma vez que os *notes* são alimentados por baterias, é importante observar o tempo de vida dessa fonte de energia. Note que um CPD de médio porte com mil bastidores implica no uso de 6 mil *notes*, considerando que três *notes* são colocados de maneira

igualmente espaçada na frente e traseira dos bastidores. O número de baterias necessárias para alimentar essa quantidade de *motes* pode ser considerável, gerando problemas que variam desde a poluição do meio-ambiente por conta do descarte de pilhas até um elevado custo de operação e manutenção da rede de sensores. Por esses motivos, nesta seção serão analisadas algumas técnicas empregadas com o intuito de gerar economia de energia.

O maior dreno de bateria que há no modelo de *mote* selecionado é a unidade de rádio-transmissão. A diferença entre envio ou recebimento de dados é desprezível, porém quando o rádio está ligado por qualquer motivo o custo é alto. Sendo assim, procurou-se minimizar tanto o envio de informações quanto o tempo de escuta.

Observou-se que a série de temperatura de vários sensores no ambiente de testes é praticamente constante e varia apenas em algumas circunstâncias, como grandes alterações na carga de trabalho, ou no longo prazo, de acordo com a temperatura externa. Sendo assim, uma tática empregada é usar um método de compressão com perdas (*lossy*) nas medidas de temperatura [34]. O mecanismo proposto consiste em atrasar o envio da medida de temperatura até que haja uma variação (positiva ou negativa) de  $\Delta$ . No entanto, caso não haja nenhuma oportunidade de envio por mais de  $t_{max}$  minutos, a medida atual é enviada. O Algoritmo 1 resume essa técnica.

---

**Algoritmo 1** Compressão de dados de temperatura

---

```

1:  $t_{envio} \leftarrow 0$ 
2:  $T_{envio} \leftarrow 0$ 
3: para cada instante  $t$  faça
4:    $T \leftarrow amostrarTemp()$  {Obtém uma amostra do sensor de temperatura.}
5:   se  $|T - T_{envio}| > \Delta$  ou  $t - t_{envio} > t_{max}$  então
6:     Envia  $\langle T, t \rangle$ 
7:      $t_{envio} \leftarrow t$ 
8:      $T_{envio} \leftarrow T$ 
9:   fim se
10: fim para

```

---

O parâmetro  $\Delta$  foi ajustado para 0,5 °C, pois julgou-se que variações inferiores a esse valor não são de interesse para as aplicações de monitoramento. A título de comparação, a precisão do hardware de sensoriamento é de 0,2 °C [32]. Através de simulações, determinou-se empiricamente que o valor de  $\Delta$  através do critério de retornos decrescentes. Através do intervalo  $t_{max}$ , pode-se detectar que um nó falhou ou não possui rota para a base (órfão) caso não haja medidas novas dentro desse período de tempo. Além disso, utiliza-se um intervalo de amostragem de 1 segundo.

Uma vez que o envio da amostra não é compulsório, o sistema fica menos exposto a

colisões de mensagens, onde dois *motes* ou mais com alcance mútuo transmitem ao mesmo tempo ocasionando uma distorção da mensagem original.

Mesmo com a diminuição da taxa de envio, o rádio ainda permanecerá ligado para escuta de mensagens, pois a qualquer momento novas mensagens podem chegar para um dado nó repassar através da rede de coleta. Para diminuir o tempo de escuta, utilizou-se um esquema de ciclo de trabalho (*duty cycle*), onde o rádio permanece ativo somente por uma fração de tempo a cada instante. Essa técnica recebe o nome de LPL – *Low-Power Listening* [37]. Mensagens que foram perdidas durante o período de sono do rádio são reenviadas. Após uma dada quantidade de falhas de reenvio, o enlace é descartado e outro é eleito pelo nó.

Tendo em vista as técnicas e protocolos empregados, experimentos indicam que o tempo de vida da rede descrita é de aproximadamente 41 dias [5].

## 3.5 Programação sem-fio

Em uma rede de milhares de sensores deve-se considerar outro aspecto prático: a instalação e manutenção de software nos sistemas embarcados. Mesmo com a utilização de simuladores [23], a rede de sensores está sujeita a *bugs* de software, necessidade de novas *features* e atualização dos seus componentes.

É possível retirar o *mote* da sua posição original, plugá-lo em um dispositivo de programação USB e depois devolvê-lo para sua posição original no *datacenter*. No entanto, realizar essa tarefa milhares de vezes é um processo lento, tedioso e passível de erros. Além disso, o sistema de monitoramento ficará inativo durante o processo a menos que cada sensor seja atualizado e reinstalado um-a-um.

Uma técnica que pode ser usada para resolver esses problemas é a chamada Programação Sem-Fio (*Over-the-Air Programming*). Utilizando a própria rede de sensores, pode-se disseminar a imagem binária contendo uma nova versão do software embarcado. Este é armazenado em uma unidade de armazenamento não-volátil (no caso do IRIS, uma memória *flash*).

O protocolo que implementa essa funcionalidade é o Deluge [8] e seu modo de disseminação é através de um *broadcast* epidêmico. Ao receber uma nova imagem do programa através da placa de programação, o *mote* reinicia e passa a enviar periodicamente uma mensagem *broadcast* contendo, entre outras informações, o número de versão da imagem

binária. Ao receber essa mensagem, cada nó verifica se a sua versão é igual ou superior à da mensagem. Não sendo o caso, significa que o nó está desatualizado e assim negocia o download da imagem. Ao terminar a instalação e reiniciar, o processo se repete e assim novos *motes* que estarão no alcance deste receberão o novo programa. Ao final do processo todos os nós possuirão a versão nova. Qualquer nó que seja colocado a partir desse ponto se integrará a rede, facilitando a substituição de nós defeituosos ou a instalação de novos dispositivos.

Através da programação sem-fio, a tarefa de atualizar o software embarcado é simples e rápida, facilitando a manutenção do sistema. Além disso, abre-se a possibilidade de desativar a rede (ou apenas parte dela) para economizar energia.

## 3.6 Ambiente de testes

A rede de monitoramento foi implantada e testada em um *datacenter* localizado no Laboratório da Pós Graduação, no Instituto de Computação. O Cluster Oscar [14] foi adquirido em conjunto com os Institutos de Computação, Física e Química e instalado em 2009. O aglomerado consiste de 42 máquinas Intel Xeon com 8 *cores* cada, dividida em 2 bastidores.

A sala em que os experimentos foram realizados pode ser vista de diversos ângulos, nas Figuras 3.2, 3.3 e 3.4, através de um modelo 3D<sup>2</sup>. A refrigeração desse ambiente é provida através de três unidades de ar-condicionado modelo *split*, posicionados sobre a saída de ar dos bastidores, como mostra a Figura 3.2.

Ao lado direito dos bastidores na Figura 3.2, há uma porta que dá acesso a sala, representada pela abertura. Devido à proximidade com os bastidores (ver Figura 3.4), a abertura da porta pode causar um desbalanceamento térmico dependendo do período em que ficar aberta.

Note que há diversos rótulos associados à *motes* produtores nas figuras. Cada rótulo identifica um nó unicamente e apresenta o conjunto de *motes* utilizados nos experimentos de detecção de anomalias da Seção 7.1. A posição espacial desses dispositivos é importante pois estão correlacionados com a região e tipo de anomalia.

---

<sup>2</sup>Note que o modelo apresentado é apenas um esboço para mostrar a posição relativa de cada objeto na sala. É importante mencionar também que a sala passou por reformas e que o modelo apresentado refere-se ao leiaute da sala na época em que os experimentos foram realizados, no primeiro semestre de 2013.

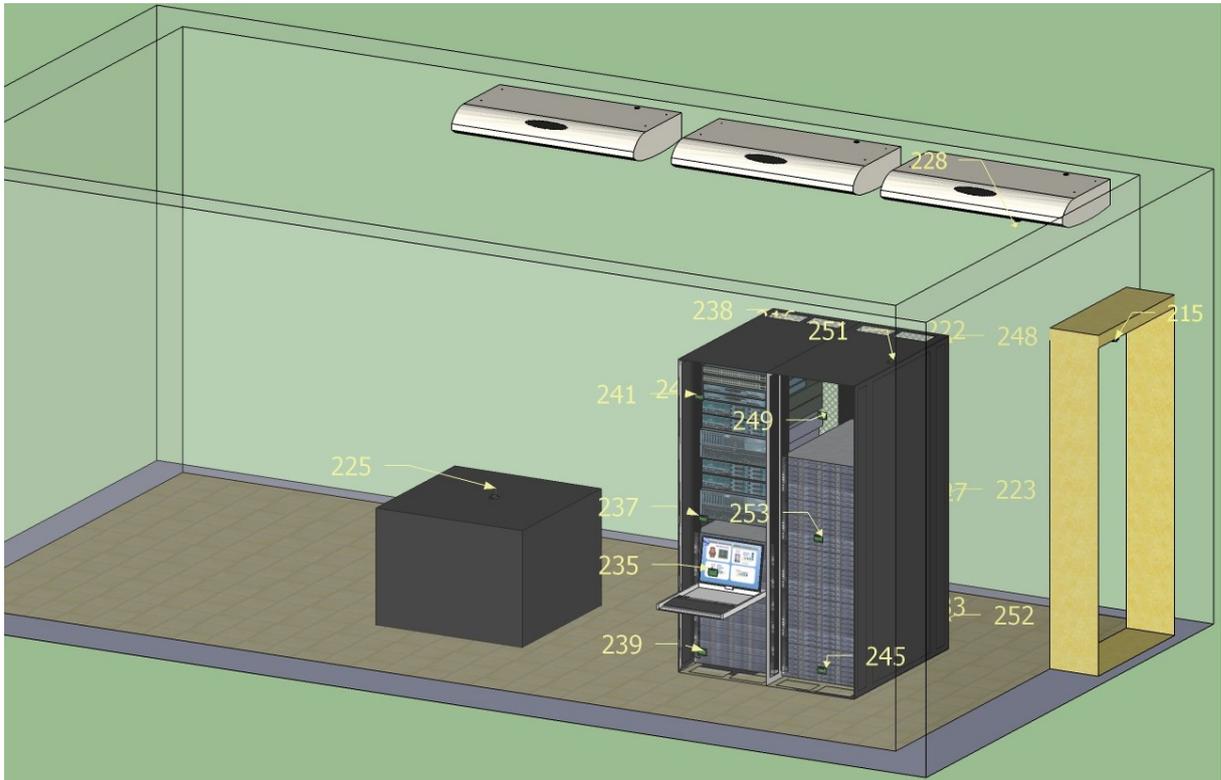


Figura 3.2: Cluster Oscar – frente

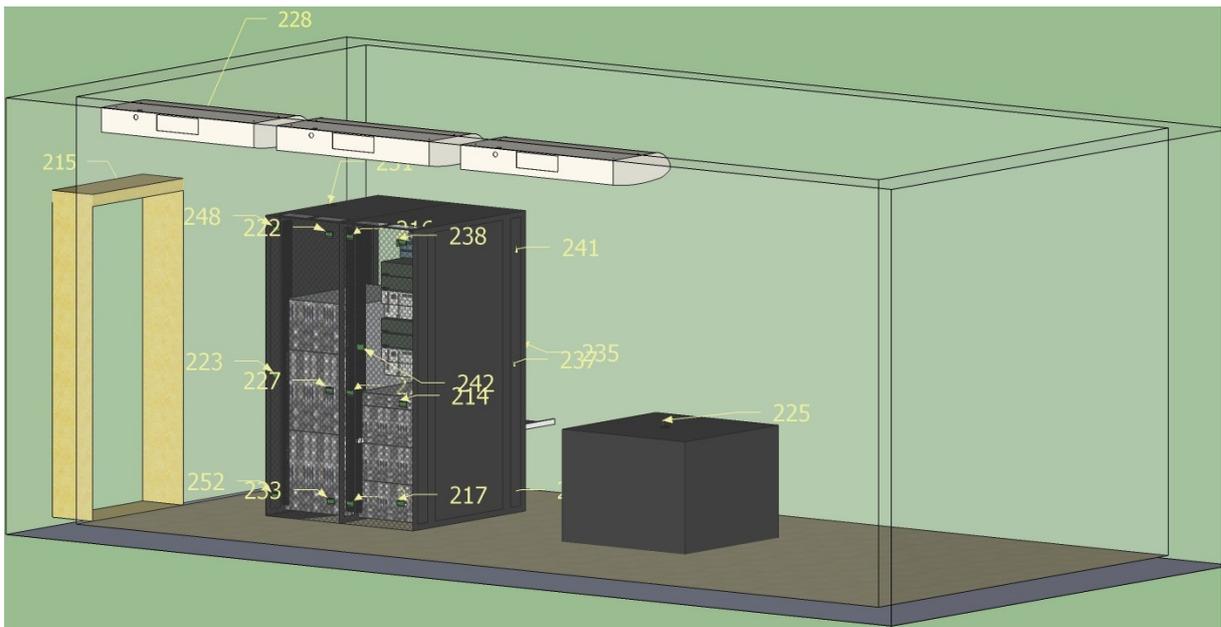


Figura 3.3: Cluster Oscar – trás

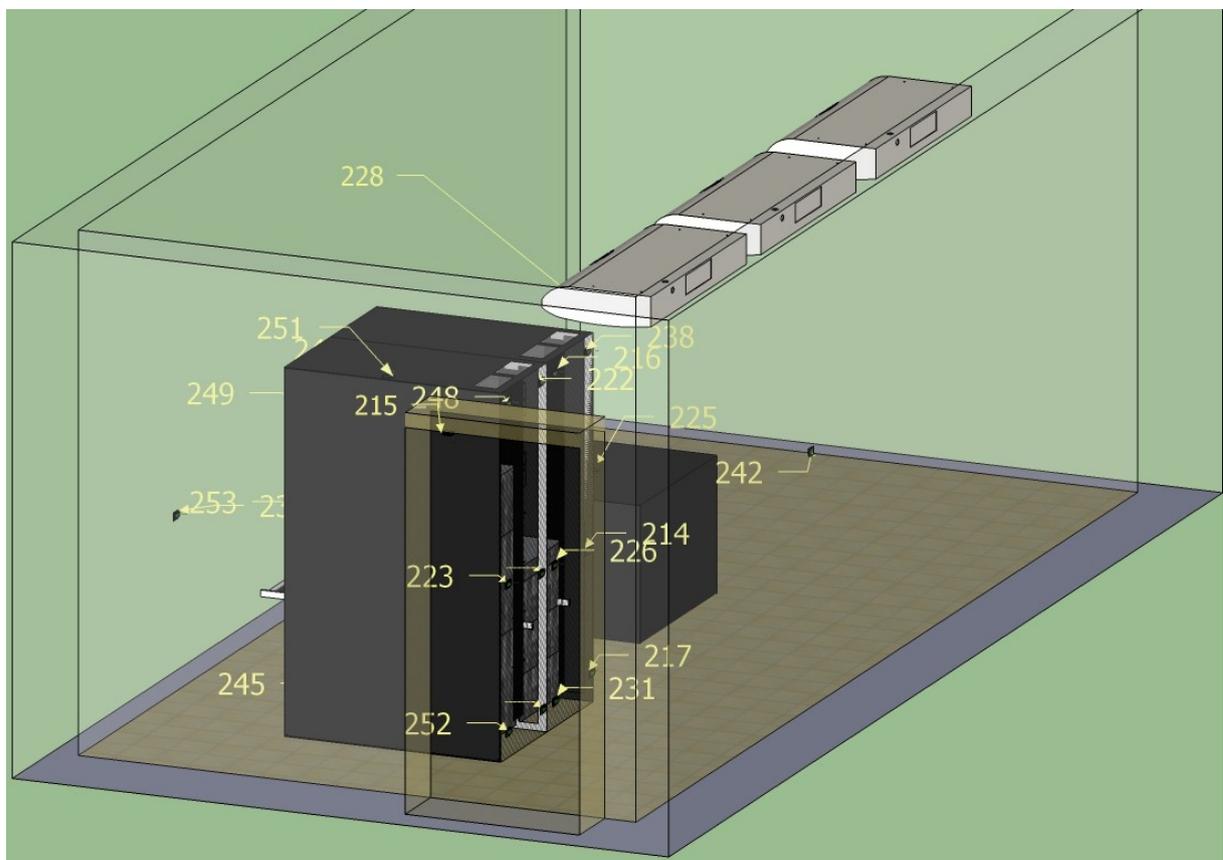


Figura 3.4: Cluster Oscar – lateral

# Capítulo 4

## Séries temporais

O foco deste capítulo é formalizar as intuições e comportamento da rede de monitoramento térmico em um arcabouço matemático e estatístico. O objetivo aqui não é fornecer uma discussão profunda sobre séries temporais, mas definir de forma clara os conceitos de apoio à proposta deste trabalho.

O conceito de séries temporais usado em Estatística e Processamento de Sinais pode ser definido como uma sequência de medidas observadas no tempo. Por “medida”, entende-se qualquer tipo de valor ou estado que se observa no tempo, como ações de uma empresa, voltagem de um circuito, distância percorrida por um veículo ou pressão de uma válvula. Geralmente essa informação é representada visualmente através de um gráfico *line chart*, como na Figura 4.1.

Séries que observam a evolução de apenas uma variável ao longo do tempo são chamadas de univariadas. Já aquelas que contemplam um conjunto de variáveis são denominadas multivariadas.

As notações e convenções utilizadas ao longo deste capítulo e em outros pontos da dissertação estão representadas na Tabela 4.1 para referência.

Tabela 4.1: Notação

Conceito	Representação	Exemplo
Valores escalares	Letra minúscula em itálicos	$a = 1$
Vetor	Letra minúscula em negrito	$\mathbf{a} = [1 \ 2 \ 3]$
Matriz	Letra maiúscula em negrito	$\mathbf{A} = \mathbf{BC}$
Matriz transposta	Superscrito da letra “T”	$\mathbf{A}^T$
Matriz inversa	Superscrito do número “-1”	$\mathbf{A}^{-1}$
Elementos de um vetor	Geralmente utilizam a mesma letra do vetor	$\mathbf{a} = [a_1 \ a_2 \ a_3]$

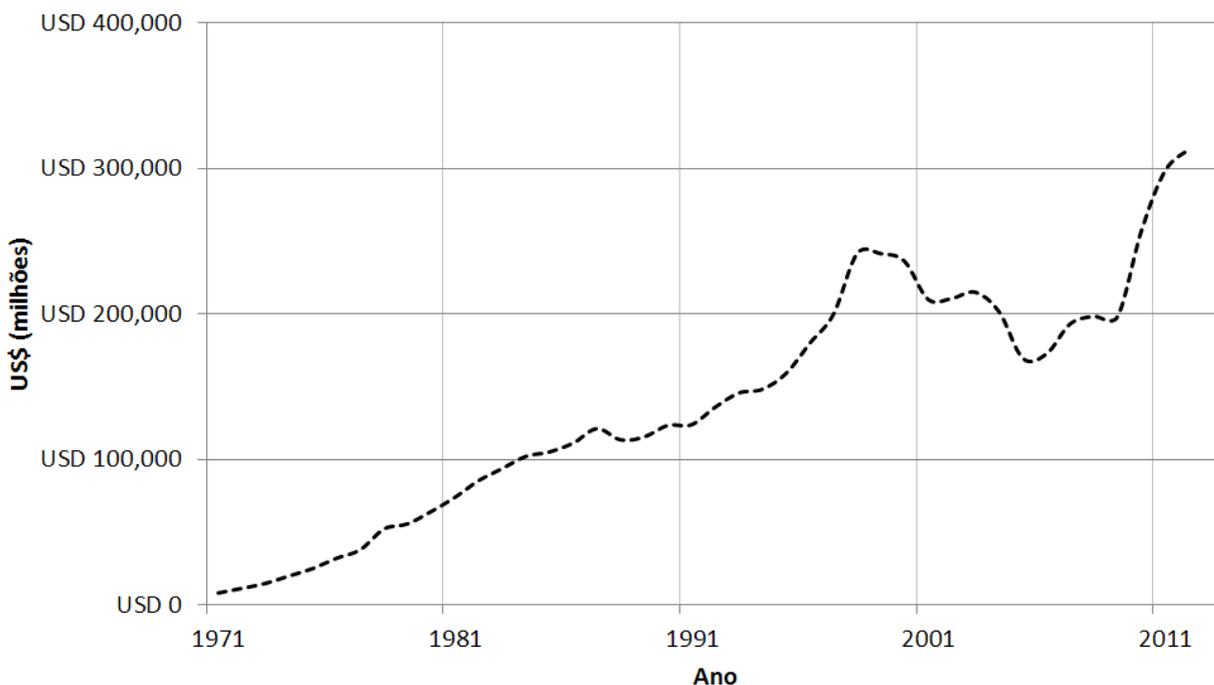


Figura 4.1: Gráfico *line chart* representando a dívida externa bruta brasileira

## 4.1 Séries temporais univariadas

Uma série temporal univariada de  $n$  amostras pode ser representada pelo vetor  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ , onde cada elemento  $x_i \in \mathbf{x}$  está ordenado de forma sequencial no tempo para  $i = 1, 2, \dots, n$ . Em outras palavras, se a medida  $x_{i-1}$  foi observada antes de  $x_i$  para qualquer valor de  $i$ . Apesar de não ser um requerimento, o intervalo de tempo (ou amostragem) dessa sequência é geralmente fixo, como é o caso deste trabalho.

## 4.2 Séries temporais multivariadas

Séries multivariadas são extensões para  $m$  dimensões do conceito apresentado anteriormente. Ao invés de um único vetor, temos uma coleção de vetores<sup>1</sup> organizados em uma matriz:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

A cada intervalo de tempo, tem-se um conjunto de  $m$  medidas, cada uma delas representando uma variável distinta em um espaço  $m$ -dimensional. No caso deste trabalho,

<sup>1</sup>Note que cada vetor  $\mathbf{x}_i$  é um vetor-coluna.

cada dimensão está associada à temperatura de um sensor localizado no *datacenter*.

A medida que  $m$  cresce para centenas ou milhares, um fenômeno chamado *curse of dimensionality* ocorre [28]. Quanto maior for o número de dimensões, maior é o volume do espaço. Consequentemente, a quantidade de dados necessária para construir uma estatística precisa cresce exponencialmente, pois os dados disponíveis tornam-se demasiadamente esparsos. Além disso, técnicas utilizadas tradicionalmente para séries univariadas geralmente tem um custo computacional proibitivo.

Para combater esses problemas, utilizam-se técnicas de redução dimensional. Métodos dessa categoria se valem de redundâncias e correlações entre as séries observadas para projetar o espaço  $m$ -dimensional em um espaço de menores dimensões, evitando o *curse of dimensionality* e diminuindo o custo computacional associado.

### 4.2.1 ACP - Análise de Componentes Principais

O ACP é um método baseado em Álgebra Linear que transforma um conjunto de variáveis correlacionadas em um conjunto de variáveis descorrelacionadas, chamadas de *componentes principais* (CP). Cada CP é construído de modo a reter a maior quantidade de informação (ou variância) possível. Por conta dessa propriedade, geralmente é possível reduzir o conjunto de CPs com perda mínima de informação em relação ao conjunto de dados original. Sendo assim, o ACP pode ser usado para reduzir a dimensionalidade do conjunto de dados [45].

A premissa de operação do ACP é que há um fenômeno subjacente à série temporal multivariada que pode ser resumido através de poucas relações lineares. Em outras palavras, assume-se que há redundância de informação no que deseja-se analisar.

O objetivo da técnica é a procura de uma nova base que re-expressa o conjunto de dados. O funcionamento do ACP começa por identificar a direção onde a variância é máxima no espaço  $m$ -dimensional. Tal processo é equivalente a ajustar uma linha de mínimos quadrados sobre os dados. Em seguida, ajusta-se uma nova linha, perpendicular à primeira, de modo a maximizar a variância nessa nova direção. O processo se repete até o  $m$ -ésimo CP, sendo que cada nova direção escolhida deve ser perpendicular às anteriores. Geometricamente, o processo é equivalente a ajustar um elipsoide sobre o conjunto de dados.

A seguir, será formalizada esta intuição utilizando propriedades de Álgebra Linear<sup>2</sup>.

---

<sup>2</sup>A demonstração que se segue foi adaptada de um tutorial sobre ACP [45].

Seja  $\mathbf{X}$  a matriz de  $m$  séries temporais e  $n$  amostras do conjunto de dados original, onde cada coluna é um vetor de amostras no instante  $t$  denominado  $\mathbf{x}_t$ . Seja  $\mathbf{Y}$  uma matriz  $m \times n$  dada pelo resultado da transformação linear  $\mathbf{Y} = \mathbf{P}\mathbf{X}$ . A matriz  $\mathbf{P}$  é ortonormal (seus vetores são unitários e perpendiculares entre si) e cada linha corresponde a um CP.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 \cdot \mathbf{x}_1 & \cdots & \mathbf{p}_1 \cdot \mathbf{x}_n \\ \vdots & \ddots & \vdots \\ \mathbf{p}_m \cdot \mathbf{x}_1 & \cdots & \mathbf{p}_m \cdot \mathbf{x}_n \end{bmatrix}.$$

Como foi dito anteriormente, cada CP é construído de modo a capturar a máxima variância em sua direção. A variância  $\sigma_{\mathbf{a}}^2$  mede a dispersão de um vetor  $\mathbf{a}$  em torno de sua média,  $\mu_{\mathbf{a}}$ . Assumindo que o conjunto de dados está centralizado, i.e.  $\mu_{\mathbf{a}} = 0$ , pode-se definir:

$$\sigma_{\mathbf{a}}^2 = \frac{1}{n-1} \sum_{i=1}^n a_i^2.$$

A covariância  $\sigma_{\mathbf{ab}}^2$  é uma estatística que mede a relação linear entre dois vetores,  $\mathbf{a}$  e  $\mathbf{b}$ . Tal medida pode ser definida como um produto escalar da seguinte maneira:

$$\sigma_{\mathbf{ab}}^2 = \frac{1}{n-1} \mathbf{ab}^T.$$

Note que a covariância de um mesmo vetor é equivalente a variância:

$$\sigma_{\mathbf{aa}}^2 = \frac{1}{n-1} \mathbf{aa}^T = \frac{1}{n-1} \sum_{i=1}^n a_i^2 = \sigma_{\mathbf{a}}^2.$$

Ao aplicar covariância em um conjunto de vetores, como  $\mathbf{X}$ , obtém-se a matriz de covariância que é definida por:

$$\mathbf{C}_{\mathbf{X}} = \frac{1}{n-1} \mathbf{X}\mathbf{X}^T.$$

Ao invés de um resultado escalar, tem-se uma matriz onde cada elemento corresponde a  $\sigma_{\mathbf{x}_i \mathbf{x}_j}^2$  onde  $i$  e  $j$  correspondem a linhas e colunas, respectivamente. Note que a diagonal de  $\mathbf{C}_{\mathbf{X}}$  contém a variância de cada vetor  $\mathbf{x}_i$ . Esta matriz informa o grau de redundância entre cada par de séries temporais que será expressada através da covariância.

Para aplicar o ACP na matriz  $\mathbf{X}$ , deseja-se que a covariância entre cada par de variáveis seja zero e variância seja máxima. Em outras palavras, a operação necessária é a diagonalização da matriz  $\mathbf{C}_{\mathbf{Y}}$ , onde:

$$\mathbf{C}_{\mathbf{Y}} = \frac{1}{n-1} \mathbf{Y}\mathbf{Y}^T = \frac{1}{n-1} \mathbf{D}$$

e  $\mathbf{D}$  é a matriz diagonal.

A fim de chegar a este resultado, expande-se  $\mathbf{Y}\mathbf{Y}^T$ :

$$\begin{aligned}\mathbf{Y}\mathbf{Y}^T &= (\mathbf{P}\mathbf{X})(\mathbf{P}\mathbf{X})^T \\ &= \mathbf{P}\mathbf{X}\mathbf{X}^T\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{X}\mathbf{X}^T)\mathbf{P}^T \\ &= \mathbf{P}\mathbf{S}\mathbf{P}^T\end{aligned}\tag{4.1}$$

A matriz  $\mathbf{S} = \mathbf{X}\mathbf{X}^T$  é simétrica, i.e.  $\mathbf{S} = \mathbf{S}^T$ . Pode-se demonstrar que qualquer matriz simétrica pode ser decomposta da seguinte forma:  $\mathbf{S} = \mathbf{E}\mathbf{D}\mathbf{E}^T$ , onde cada coluna de  $\mathbf{E}$  corresponde aos autovetores de  $\mathbf{S}$ .

Neste ponto é necessário construir  $\mathbf{P}$  de forma que  $\mathbf{Y}\mathbf{Y}^T = \mathbf{D}$ . Define-se  $\mathbf{P} = \mathbf{E}^T$ . Isso é o mesmo que dizer que  $\mathbf{P}$  é uma matriz onde cada linha corresponde a um autovetor de  $\mathbf{S}$ . Dado que  $\mathbf{P}$  é uma matriz ortogonal, sabe-se que  $\mathbf{P}^T = \mathbf{P}^{-1}$ . Utilizando estas propriedades, mostra-se que a escolha de  $\mathbf{P}$  de fato diagonaliza  $\mathbf{Y}\mathbf{Y}^T$ :

$$\begin{aligned}\mathbf{Y}\mathbf{Y}^T &= \mathbf{P}\mathbf{S}\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{E}\mathbf{D}\mathbf{E}^T)\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{P}^T\mathbf{D}\mathbf{P})\mathbf{P}^T \\ &= (\mathbf{P}\mathbf{P}^T)\mathbf{D}(\mathbf{P}\mathbf{P}^T) \\ &= (\mathbf{P}\mathbf{P}^{-1})\mathbf{D}(\mathbf{P}\mathbf{P}^{-1}) \\ &= \mathbf{D}\end{aligned}\tag{4.2}$$

A seguir, resume-se os passos necessários para o cálculo do ACP através do Processo 1.

---

#### Processo 1 Cálculo do ACP

1. Centralizar cada vetor de  $\mathbf{X}$  subtraindo a sua média.
  2. Calcular os autovetores de  $\mathbf{X}\mathbf{X}^T$  de modo a construir a matriz  $\mathbf{P}$ .
  3. Calcular a transformação  $\mathbf{Y} = \mathbf{P}\mathbf{X}$ .
-

### 4.2.2 SPIRIT

A seção anterior revelou uma ferramenta útil para a redução de dimensionalidade e processamento de séries temporais multivariadas, o ACP. No entanto, deseja-se realizar o processamento dessas séries em tempo real (*online*). A cada instante de tempo uma nova medida produzida pelos sensores deve ser incorporada para as aplicações de previsão de temperatura e detecção de anomalias. Contudo, o ACP assume que todos os dados estão presentes *a priori*, o que é impossível no cenário proposto.

Uma alternativa ingênua seria aplicar repetidamente o ACP a cada nova medida que chega ao sistema. Armazenar todos os dados em memória e processar cada vez mais informação é inviável na prática.

É necessário utilizar uma abordagem que seja capaz de calcular iterativamente o ACP, sem que os requisitos de memória e processamento dependam de  $t$ . Um algoritmo que cumpre essa tarefa é o SPIRIT – *Streaming Pattern dIscoveRy in multIple Time-series* [40].

#### 4.2.2.1 Definição

O SPIRIT utiliza uma definição análoga ao ACP, onde  $\mathbf{Z} = \mathbf{W}\mathbf{X}$ . A matriz  $\mathbf{X}$  tem a mesma definição anterior: trata-se das séries temporais de entrada. A matriz  $\mathbf{W}$ , denominada *weights* (pesos), corresponde aos componentes principais ( $\mathbf{P}$ ). Já os elementos de  $\mathbf{Z}$ , chamados de *variáveis ocultas*, são aproximações das projeções na nova base definida por  $\mathbf{W}$ .

O funcionamento do SPIRIT é baseado em uma técnica chamada *Recursive Least Squares* (RLS) [17]. Através dela, o SPIRIT estima iterativamente as direções e valores dos componentes principais.

Um conceito importante para o SPIRIT é a *energia*, original da área de Processamento de Sinais. Ela pode ser definida para um vetor  $\mathbf{a}$  como:

$$E_t(\mathbf{a}) = \frac{1}{t} \sum_{i=1}^t \|\mathbf{a}_i\|^2.$$

O jargão da área de Processamento de Sinais frequentemente emprega termos físicos, como “potência” e “energia”. “Energia” é um termo usado para expressar a área sob a curva do sinal de potência, ou no caso discreto, o somatório de cada amostra (supondo taxa de amostragem com valor 1).

O funcionamento do SPIRIT se baseia no rastreamento dos componentes principais. O algoritmo mede o erro de reconstrução e refina sua estimativa iterativamente. A variável de energia, como uma maneira de sumarizar a magnitude de um vetor, pode ser usada para medir a discrepância entre a série multivariada original e aquela definida pelo SPIRIT. A energia de uma variável oculta,  $E_t(\mathbf{y}_i)$ , é usada para guiar a atualização de cada peso  $\mathbf{w}_i$  de forma que quanto menor for a energia, maior será a correção feita no componente principal.

O funcionamento do algoritmo de atualização de pesos está descrito no Algoritmo 2. O primeiro passo consiste em iniciar a matriz  $\mathbf{W}$  com uma matriz ortonormal, que no caso é a base canônica (*standard basis*). Em seguida, inicia-se o vetor  $\mathbf{d}$  com um valor pequeno, porém maior que zero. Isso é feito para que não ocorra uma divisão por zero na etapa 9.

---

**Algoritmo 2** Atualização dos pesos (aproximação dos CPs)

---

- 1: Inicie  $\mathbf{w}_i$  com vetores unitários, i.e.,  $\mathbf{w}_1 = [1 \ 0 \ \dots \ 0]^T$ ,  $\mathbf{w}_2 = [0 \ 1 \ \dots \ 0]^T$ ,  $\dots$ ,  $\mathbf{w}_k = [0 \ 0 \ \dots \ 1 \ \dots \ 0]^T$ .
  - 2: Inicie os elementos de  $\mathbf{d}$  com um valor pequeno, positivo (como  $10^{-2}$ ).
  - 3:  $\hat{\mathbf{x}} \leftarrow \mathbf{x}_t$  {Faz uma cópia do vetor de entrada.}
  - 4: **para** cada instante  $t$  **faça**
  - 5:   **para**  $1 \leq i \leq k$  **faça**
  - 6:      $z_i \leftarrow \mathbf{w}_i^T \hat{\mathbf{x}}$  {Realiza a projeção sobre  $\mathbf{W}$ .}
  - 7:      $d_i \leftarrow \lambda d_i + z_i^2$  {Atualiza a energia da  $i$ -ésima projeção.}
  - 8:      $\mathbf{e}_i \leftarrow \hat{\mathbf{x}} - z_i \mathbf{w}_i$  {Atualiza o vetor do erro de reconstrução.}
  - 9:      $\mathbf{w}_i \leftarrow \mathbf{w}_i + \frac{1}{d_i} z_i \mathbf{e}_i$  {A correção do CP é inversamente proporcional a energia da variável oculta.}
  - 10:    $\hat{\mathbf{x}}_{i+1} \leftarrow \hat{\mathbf{x}}_i - z_i \mathbf{w}_i$  {O processo se repete com o restante de  $\mathbf{x}_t$ .}
  - 11:   **fim para**
  - 12: **fim para**
- 

Em seguida, realiza-se a projeção (passo 6), atualiza-se a energia daquela variável (passo 7) e mede-se o erro obtido (passo 8). Note que o erro é calculado em relação a  $\hat{\mathbf{x}}$  e não  $\mathbf{x}_t$ . Isso ocorre pois cada variável oculta subsequente irá descrever os resíduos entre a projeção atual e a anterior (veja o passo 10). Os pesos são atualizados na etapa 9, de maneira inversamente proporcional a energia. O parâmetro  $\lambda$  será explicado posteriormente e pode ser considerado como uma constante igual a 1, por enquanto.

A complexidade deste algoritmo depende apenas de  $k$  e  $n$ , ou seja, o número de variáveis ocultas e o tamanho de cada vetor. Assume-se que há um grau de correlação entre as séries temporais, tal que  $k \ll n$ . Sendo assim, o custo computacional (em operações aritméticas) e o de armazenagem (em função do número de variáveis de ponto-

flutuante) é  $O(nk)$ .

#### 4.2.2.2 Variação dinâmica do número de variáveis ocultas

No SPIRIT, o valor de  $k$  flutua de maneira dinâmica em função do erro de reconstrução obtido. Lembre-se que o ACP é um método de redução dimensional e que, por construção, cada variável do sistema agrega a maior quantidade de informação (no sentido de minimizar o quadrado dos resíduos) possível. Uma consequência prática disso é que frequentemente pode-se descartar muitas dessas variáveis sem incorrer em um erro proibitivo. No entanto, o erro só é zero quando  $k = n$ .

A técnica adotada para realizar este controle pelos autores do SPIRIT é a chamada *energy-thresholding*. Nesse método, define-se um limiar superior,  $max_e$ , e inferior,  $min_e$ , da razão de energia capturada pelo SPIRIT em relação às variáveis originais,  $E(\mathbf{y}_t)/E(\mathbf{x}_t)$ . Quando essa razão for menor que  $min_e$ , significa que o número de variáveis ocultas já representa com precisão suficiente o sistema neste momento, e assim diminui-se o número de variáveis ocultas. De maneira similar, caso esse valor ultrapasse  $max_e$ , uma nova variável é introduzida para diminuir o erro.

O Algoritmo 3 sintetiza os passos realizados para a variação de  $k$ . Note que esses passos são executados após o *loop* interno do Algoritmo 2, a cada instante de tempo  $t$ .

---

#### Algoritmo 3 Ajuste de $k$

---

```

1:  $EX \leftarrow \lambda \cdot EX + E(\mathbf{x}_t)$ 
2:  $EY \leftarrow \lambda \cdot EY + E(\mathbf{y}_t)$ 
3:  $r \leftarrow EY/EX$ 
4: se  $r < min_e$  então
5:    $k \leftarrow max(k - 1, 1)$ 
6: senão
7:   se  $r > max_e$  então
8:      $k \leftarrow k + 1$ 
9:   fim se
10: fim se

```

---

O Algoritmo 3 possui uma propriedade importante que permite garantir limiares para o erro máximo e mínimo de reconstrução. Formalmente, é possível mostrar que:

$$1 - max_e \leq \sum_{i=1}^t \frac{\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|^2}{\|\mathbf{x}_t\|^2} \leq 1 - min_e.$$

Nessa equação,  $\tilde{\mathbf{x}}_t$  representa a reconstrução das  $n$  séries temporais, equivalente a  $\mathbf{z}_t \mathbf{w}_t$ . A fração no meio da desigualdade representa o erro quadrado relativo e seu valor está no

intervalo  $[0; 1]$ . É intuitivo estipular os limites para esse erro. Por exemplo, para manter o erro no intervalo  $(10\%, 20\%)$ , basta estipular  $max_e = 90\%$  e  $min_e = 80\%$ . A demonstração desse teorema pode ser encontrada no artigo original [40].

Uma limitação do esquema representado até agora é que assume-se um processo estocástico estacionário. Nele, não é previsto que a média e variância do sistema varie com o tempo. No entanto é muito comum que haja uma flutuação dessas estatísticas em séries temporais que refletem um sistema físico, como o monitoramento térmico. Pensando nisso, o RLS utiliza um “fator de esquecimento”,  $\lambda$ , que permite considerar o comportamento recente das séries temporais com mais intensidade do que o passado.

Trata-se de um valor no intervalo  $[0; 1]$ , com valores típicos na faixa  $[0,96; 0,98]$ . O fator de esquecimento segue um decaimento exponencial, onde valores mais próximos de 0 reduzem valores históricos mais rapidamente do que valores próximos a 1.

O fator  $\lambda$  está associado às variáveis de energia no Algoritmo 2 e 3, pois deseja-se considerar apenas a energia calculada mais recentemente para a adaptação da estimativa dos componentes principais. Por razão de consistência, também é necessário alterar o valor da estimativa de energia no Algoritmo 3.

#### 4.2.2.3 Centralização das séries

Como foi visto anteriormente o primeiro passo do Processo 1 é a centralização de cada vetor  $\mathbf{x}_i$ . No entanto, a média não é conhecida *a priori* e deve ser estimada em tempo real. Para tanto, basta observar a seguinte relação da fórmula da média aritmética [20]:

$$\bar{x}_n = \frac{(n-1)\bar{x}_{n-1} + x_n}{n} = \frac{n\bar{x}_{n-1} + x_n - \bar{x}_{n-1}}{n} = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}$$

Logo, o algoritmo necessário para atualizar a média com o novo valor  $v$  de um dado vetor  $x_i$  é simplesmente:

$$\bar{\mathbf{x}}_i \leftarrow \bar{\mathbf{x}}_i + \frac{v - \bar{\mathbf{x}}_{n-1}}{n}.$$

# Capítulo 5

## Detecção de Anomalias

A Detecção de Anomalias é uma tarefa de Mineração de Dados que tem como objetivo a identificação de padrões inesperados em um conjunto de dados. Neste trabalho, o conjunto de dados em questão é uma série temporal multivariada de temperatura e, portanto, a discussão de Detecção de Anomalias se limitará à séries temporais.

Apesar do termo “anomalia”, técnicas de Detecção de Anomalias também são utilizadas para procurar por *padrões* surpreendentes, inesperados ou curiosos, dependendo do contexto em que a técnica é aplicada [28].

Considere o problema de identificar doenças cardíacas através de um eletrocardiograma. A Figura 5.1 mostra um eletrocardiograma humano com a presença de uma arritmia. Os quatro primeiros batimentos representam um padrão saudável do nó sinusal (estrutura responsável pela regulação do ritmo do coração). No entanto, após um período de omissão de um batimento, indicado pela seta cinza, o quinto batimento apresenta um formato diferente dos anteriores como mostra a seta pontilhada. Esse exemplo serve para ilustrar um caso onde a descoberta de uma anomalia poderia ser útil para alertar um profissional de saúde para um indício de uma doença.

A definição exata do que constitui uma anomalia varia bastante na literatura. Em alguns casos, anomalias se aproximam da definição de *outliers* usada em Estatística: itens de um conjunto de dados que diferem dos demais devido à ruídos ou imprecisões no processo de medição. Nesse caso, anomalias são identificadas e removidas com o intuito de limpar o conjunto de dados para extrair estatísticas com maior relevância.

Um outro tipo de definição se alinha com a mineração de padrões frequentes em séries temporais, chamados de *motifs*. Segundo essa definição, anomalias são “padrões cuja ocorrência difere significativamente da esperada, baseado no histórico de dados” [19].

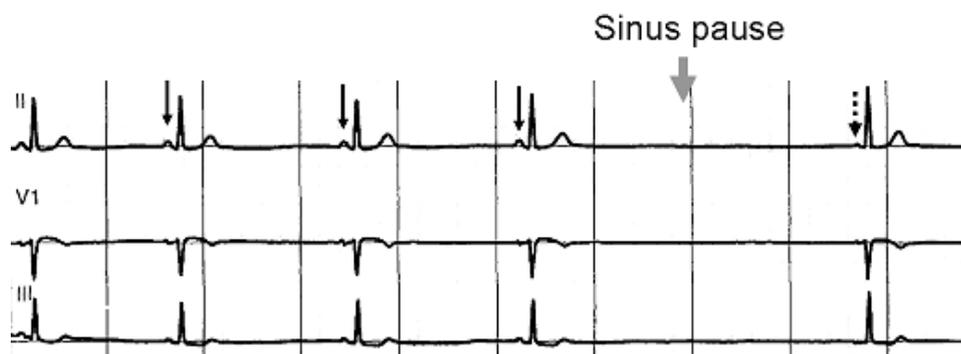


Figura 5.1: Eletrocardiograma com a presença de uma arritmia

Sendo assim, a tarefa de detecção de anomalias se resume a identificar padrões e analisar a sua frequência durante a evolução da série temporal.

Técnicas da família de detecção de anomalias podem ser classificadas em duas categorias: modelos supervisionados e não-supervisionados. No primeiro caso, é preciso prover uma série temporal rotulada com regiões anômalas para treinar um modelo que aprenderá a classificar entre dois estados: normal e anômalo. Já a segunda categoria gera um modelo que adere ao processo observado. Desvios do modelo são encarados como anomalias, uma vez que assume-se que a maior parte da série temporal é livre de anomalias e que o modelo descreve o processo de forma satisfatória.

Há aplicações em que anomalias são esperadas com uma certa frequência e de modo estereotipado o suficiente de forma a viabilizar uma base de treinamento rotulada. Um exemplo desse tipo de aplicação são ataques de negação de serviço e detecção de fraudes [46]. Em outros casos, anomalias são eventos raros e variados que não podem ser agrupados sob um molde pré-estabelecido e portanto só podem ser tratados por técnicas não-supervisionadas.

Neste trabalho, deseja-se empregar a Detecção de Anomalias para capturar eventos térmicos que podem estar relacionados a ineficiências térmicas e ilhas de calor. De posse dessa informação, o operador do *datacenter* será capaz de rastrear e identificar as causas dessas anomalias gerando economia de energia e um ambiente mais estável para suas aplicações.

Ao gerar indicadores de anomalia, o sistema está evitando que o operador analise manualmente uma miríade de dados que tornaria o monitoramento em tempo real impraticável. Cada anomalia detectada é um alerta que deve ser inspecionado e validado pelo operador através de câmeras, monitoramento de carga e outras ferramentas que podem estar disponíveis no *datacenter*.

De posse dessa informação, é possível gerar soluções que variam desde a mudança de leiaute do CPD, até o balanceamento de carga nos servidores, a fim de proteger o equipamento e diminuir os custos energéticos de operação.

As anomalias esperadas neste trabalho podem ter as mais diversas fontes e, consequentemente, perfis térmicos com grande variabilidade. O mal funcionamento de uma ventoinha de um servidor pode aquece-lo a ponto de gerar uma ilha de calor numa dada região do *datacenter*, porém sem causar risco de um superaquecimento. A porta de uma sala de servidores pode ficar aberta por um tempo considerável durante a manutenção ou instalação de equipamentos, gerando uma ineficiência térmica. Existe um caso registrado [3] onde a obstruções da saída de ar de um piso perfurado causou um desbalanceamento térmico na região. Dada a grande variabilidade de casos, deve-se utilizar uma técnica não-supervisionada.

## 5.1 Detecção de anomalias com o SPIRIT

O SPIRIT é uma técnica para a estimação em tempo real dos componentes principais de uma série temporal multivariada. No Capítulo 4 mostrou-se que o controle do número de variáveis ocultas garante um limiar inferior e superior para o erro de reconstrução.

Quando o erro aumenta por algum motivo, significa que o conjunto de variáveis utilizadas para representar o fenômeno observado não é suficiente e assim é introduzida uma dimensão extra com a finalidade de aumentar a representatividade do sistema. Esse comportamento pode ser visto sobre a ótica do processo de Detecção de Anomalias como um comportamento anômalo. Analogamente, pode-se fazer a mesma observação para o caso em que o número de dimensões decai.

Utilizando essa intuição, pode-se definir uma anomalia como uma mudança *significativa* na correlação das séries observadas. A definição do que configura uma mudança significativa é dada pela taxa de erro desejada (em função de  $min_e$  e  $max_e$ ).

O Algoritmo 4 descreve o processo de geração de alarmes. O algoritmo compara o número de variáveis ocultas atual contra o anterior a cada instante de tempo. Caso haja um aumento do número de variáveis, um alarme é gerado. Na prática, é comum que haja um aumento de  $k$  seguido de uma diminuição do número de variáveis ocultas quando o processo se estabiliza. Sendo assim, a ultrapassagem de  $min_e$  geralmente está associada a um aumento anterior de  $k$  e portanto se refere a mesma anomalia. Por essa razão, o algoritmo proposto considera apenas variações positivas de  $k$ .

---

**Algoritmo 4** Detecção de anomalias com SPIRIT

---

```
1:  $t_{\text{alarme}} \leftarrow t$ 
2:  $k_{\text{anterior}} \leftarrow 0$ 
3:  $\Delta \leftarrow 60$  {Tempo em segundos.}
4: para cada instante  $t$  faça
5:   se  $k > k_{\text{anterior}}$  e  $t_{\text{alarme}} > \Delta$  então
6:     Gera um alarme no tempo  $t$ 
7:      $t_{\text{alarme}} \leftarrow t$ 
8:   fim se
9:    $k_{\text{anterior}} \leftarrow k$ 
10: fim para
```

---

O parâmetro  $\Delta$ , na linha 5, tem como objetivo limitar a geração de múltiplos alarmes sobre a mesma anomalia, pois há casos onde é necessário incrementar o número de  $k$  sequencialmente até que o erro diminua para a faixa esperada.

O trecho interno ao *loop* do Algoritmo 4 é executado logo após o Algoritmo 3, que atualiza o parâmetro  $k$ . Sendo assim, a complexidade da proposta não é alterada.

# Capítulo 6

## Previsão de superaquecimentos

Neste capítulo serão apresentados os métodos selecionados para realizar a previsão de superaquecimentos. Foram escolhidos previsores de três categorias: Suavização Exponencial, Autoregressão e Perceptrons. Enquanto este capítulo está focado somente em apresentar esses métodos, no Capítulo 7 é realizada uma série de experimentos para definir qual o previsor é mais adequado para esta aplicação.

A previsão térmica tem o objetivo de auxiliar em técnicas de escalonamento de carga em outros trabalhos [35, 36]. Através do conhecimento do perfil térmico, pode-se balancear a carga de forma a reduzir o desbalanceamento térmico de servidores e assim economizar energia de resfriamento dos CRACs.

No entanto, este trabalho segue um objetivo diferente: prevenir contra superaquecimentos no CPD. A medida que soluções de economia de energia como consolidação de carga através de máquinas virtuais ganham maior adoção [42], a margem de segurança de operação fica cada vez mais próxima. Sendo assim, necessita-se de um método pró-ativo, que avise aos operadores com antecedência que há um risco iminente de superaquecimento no *datacenter*. Através desse mecanismo, espera-se que os operadores consigam tempo suficiente para atuar no sistema, como distribuir carga para um outro conjunto de máquinas, longe do foco do problema.

### 6.1 Naïve

O previsor Naïve (ingênuo) é apresentado aqui apenas com o objetivo de comparação com outras técnicas. Seu funcionamento consiste em simplesmente projetar a medida observada no instante atual,  $t$ , para  $w$  segundos no futuro ( $t + w$ ). A Equação 6.1 mostra

seu funcionamento, onde  $p_{t+w}$  representa a previsão para o tempo  $t + w$ .

$$p_{t+w} = x_t \quad (6.1)$$

## 6.2 Suavização Exponencial

Esta seção trata de métodos de suavização que além de removerem ruído da série temporal também servem para realizar previsão [29].

### 6.2.1 Média Exponencial

Este método funciona atribuindo pesos cada vez menores para dados do passado enquanto incorpora novas medidas. A Equação 6.2 mostra o funcionamento do método.

$$\begin{aligned} p_1 &= x_t \\ p_{t+1} &= \alpha \cdot x_t + (1 - \alpha) \cdot p_t \end{aligned} \quad (6.2)$$

O parâmetro  $\alpha$  é conhecido como *fator de suavização* e é uma constante real no intervalo  $[0; 1]$ . Quando próximo de 1, as medidas recentes são levadas em conta com maior intensidade. Já valores próximos a zero privilegiam o histórico capturado até o momento.

Cada novo valor da série é incorporado no método enquanto os valores antigos são atenuados de maneira exponencial em uma grande média ponderada. A Equação 6.4 e subsequentes mostram esse efeito. Note que cada termo  $x_t$  tem um peso proporcional a  $(1 - \alpha)^t$  no somatório.

$$p_{t+1} = \alpha \cdot x_t + (1 - \alpha) \cdot p_t \quad (6.3)$$

$$= \alpha \cdot x_t + (1 - \alpha) \cdot [\alpha \cdot x_{t-1} + (1 - \alpha) \cdot p_{t-1}] \quad (6.4)$$

$$= \alpha \cdot [x_t + (1 - \alpha) \cdot x_{t-1}] + (1 - \alpha)^2 \cdot p_{t-1} \quad (6.5)$$

$$= \alpha \cdot [x_t + (1 - \alpha) \cdot x_{t-1} + (1 - \alpha)^2 \cdot x_{t-2} + \dots] + (1 - \alpha)^n \cdot p_1 \quad (6.6)$$

Uma técnica tradicional, a Média Móvel, também é capaz de atribuir diferentes pesos aos termos mais recentes. No entanto, uma desvantagem desse modelo é a necessidade de armazenar uma quantidade de itens igual ao tamanho da janela da média, o que não

ocorre na Média Exponencial. Em ambos os casos, o custo para atualizar a previsão é constante e independe de  $t$ .

### 6.2.2 Método Linear de Holt

A Média Exponencial funciona bem para modelar processos estacionários que não possuem tendências (i.e. a média e variância são constantes). A fim de melhorar a eficácia desse método foi criado o Método Linear de Holt (ou simplesmente Holt), também chamado de Média Exponencial Dupla [29]. Essa técnica utiliza um componente extra para capturar as tendências positivas ou negativas na série temporal. As Equações 6.7–6.11 mostram a sua inicialização e funcionamento.

$$L_2 = x_1 \quad (6.7)$$

$$b_2 = x_2 - x_1 \quad (6.8)$$

$$L_t = \alpha \cdot x_t + (1 - \alpha) \cdot (L_{t-1} + b_{t-1}) \quad (6.9)$$

$$b_t = \beta \cdot (L_t - L_{t-1}) + (1 - \beta) \cdot b_{t-1} \quad (6.10)$$

$$p_{t+w} = L_t + w \cdot b_t \quad (6.11)$$

O parâmetro  $L_t$  representa a estimativa para o coeficiente linear (*level*), enquanto  $b_t$  o coeficiente angular (reta tangente) da série temporal no tempo  $t$ . O valor de  $L_t$  é calculado através da Equação 6.9, que consiste em realizar uma média exponencial entre o valor observado atualmente e aquele previamente estimado. Na Equação 6.10, o coeficiente angular atual é estimado através da diferença entre os parâmetros  $L_t$  e  $L_{t-1}$  e é realizada uma média exponencial entre esse valor e o anterior,  $b_{t-1}$ . Esse comportamento tem o intuito de suavizar algum ruído restante que haja nas medidas  $L_t$ . Por fim, o valor previsto é calculado de maneira linear entre  $L_t$  e  $b_t$ , como mostra a Equação 6.11.

Uma vez que as séries temporais utilizadas neste trabalho não são estacionárias, escolheu-se o Holt para representar esta família de técnicas de previsão. Apesar de existirem técnicas capazes de capturar a sazonalidade de séries temporais, elas não serão abordadas neste trabalho. Mesmo que hajam comportamentos periódicos na série temporal, deseja-se prever apenas casos que podem levar a superaquecimentos de uma região do CPD. Sendo assim, esse comportamento pode ser modelado através de uma curva linear de temperatura.

## 6.3 Autoregressão

Um modelo simples para previsão de séries temporais é a chamada Regressão Linear. Nesse modelo, a previsão é realizada através da combinação linear de  $q$  variáveis latentes  $a_i$  dada por:

$$p = \phi_0 + \phi_1 \cdot a_1 + \phi_2 \cdot a_2 + \dots + \phi_p \cdot a_q + e,$$

onde  $\phi_j$  representa um parâmetro a ser determinado e  $e$  o ruído não mapeado pelo modelo.

Pode-se utilizar os valores observados anteriormente na série temporal no lugar das variáveis latentes. Nesse caso, trata-se de uma regressão com a própria série temporal, que recebe o nome de Autoregressão e é representada por:

$$p_t = \phi_0 + \phi_1 \cdot x_{t-1} + \phi_2 \cdot x_{t-2} + \dots + \phi_p \cdot x_{t-q} + e_t.$$

A ordem do modelo Autoregressão é dada pela quantidade de variáveis empregadas no modelo e é representada pela seguinte notação: AR( $q$ ).

O modelo AR é um caso específico do modelo mais geral, chamado ARIMA (*Autoregressive Integrated Moving Average*), onde os componentes de integração e média móvel não são usados. A metodologia Box-Jenkins permite selecionar o modelo mais indicado para cada série temporal através de um conjunto de análises. Nesta dissertação, no entanto, utiliza-se um modelo AR(2) que foi selecionado previamente em um outro trabalho [25] como apropriado para previsão de séries temporais de temperatura em um *datacenter*. O modelo de previsão para o AR(2) está sumarizado abaixo:

$$p_{t+1} = \phi_0 + \phi_1 \cdot x_t + \phi_2 \cdot x_t + e_{t+1}.$$

A fórmula anterior contém o termo  $e_{t+1}$ , que é desconhecido e é tomado como zero [29]. Além disso, observe que a previsão em questão é realizada somente um passo a frente (i.e.,  $p_{t+1}$ ). Porém, é possível realizar previsão para intervalos mais longos se utilizarmos o próprio valor estimado no instante anterior como entrada para a próxima previsão [29], como nas Equações 6.12 e 6.13.

$$p_{t+1} = \phi_0 + \phi_1 \cdot x_t + \phi_2 \cdot x_t \tag{6.12}$$

$$p_{t+w} = \phi_0 + \phi_1 \cdot p_{t+w-1} + \phi_2 \cdot p_{t+w-2} \tag{6.13}$$

## 6.4 Perceptrons

Perceptrons são um tipo simplista de Rede Neural e podem ser usados para diversas tarefas, como classificação e regressão. Neste trabalho, o interesse é em algoritmos de previsão da categoria *online machine learning*, que são usados para realizar o aprendizado de maneira incremental ao invés de um treinamento sobre um *dataset* fixo.

Os Perceptrons desta seção têm a função de ativação dada pela Equação 6.14. Note que há apenas um parâmetro de entrada,  $x_t$ , associado ao peso  $r_t$ .

$$p_{t+w} = r_t \cdot x_t \quad (6.14)$$

Redes mais complexas foram utilizadas com sucesso em um outro trabalho para previsão do perfil térmico [35]. No entanto, o escopo deste trabalho não se resume a previsão térmica e, portanto, apresentaremos apenas técnicas mais simples.

### 6.4.1 Perceptron Linear

O primeiro Perceptron que será analisado é o chamado Perceptron Linear (ou LinearPerceptron). Seu objetivo é realizar a previsão tomando como entrada a temperatura do instante atual. Dado que o aprendizado é supervisionado, a entrada tem a forma  $\langle x_t \rangle$  e a saída esperada  $\langle x_{t+w} \rangle$ .

O ajuste do peso  $r_t$  é dado pela regra delta (*delta rule*), com intensidade regulada pelo parâmetro  $L$  (taxa de aprendizado).

$$r_t = L \cdot (x_t - p_t) \cdot x_{t-w} \quad (6.15)$$

### 6.4.2 Perceptron Linear com Rolamento

Um problema com o perceptron anterior é que é difícil realizar o aprendizado utilizando uma saída tão distante do tempo  $t$ , como  $\langle x_{t+w} \rangle$ . É mais fácil construir um previsor para o instante  $t + 1$  do que  $t + w$ . Utilizando a mesma abordagem da Seção 6.3, pode-se criar uma espécie de “rolamento”, onde as previsões anteriores são usadas para alimentar a função de ativação do perceptron, como mostra a Equação 6.16. De forma a diferenciar este perceptron do LinearPerceptron, utiliza-se o nome RollingPerceptron.

$$p_{t+w} = r_t \cdot p_{t+w-1} \quad (6.16)$$

$$= r_t \cdot r_t \cdot p_{t+w-2} \quad (6.17)$$

$$= r_t^w \cdot x_t \quad (6.18)$$

## 6.5 Erros

Qualquer técnica de previsão está sujeita a erros, que são dados pela diferença entre a previsão feita ( $p_t$ ) e a observação ( $x_t$ ) do momento. A Equação 6.19 formaliza essa relação.

$$e_t = p_t - x_t \quad (6.19)$$

Uma ideia simples seria usar a equação anterior para medir o erro médio (ME – *Mean Error*) obtido em uma série temporal, dando origem à Equação 6.20.

$$ME = \frac{1}{N} \cdot \sum_{t=1}^N e_t \quad (6.20)$$

No entanto, tal métrica mede apenas o viés acumulado dos erros (i.e. se o previsor subestima ou superestima as observações). Isso ocorre pois medidas de  $e_t$  tendem a se cancelar. Sendo assim, essa métrica é de utilidade limitada.

Existem diversas estatísticas que podem ser usadas para sumarizar o erro em um experimento. Talvez a mais usada seja o MSE – *Mean Square Error*, que é dada pela Equação 6.21.

$$MSE = \frac{1}{N} \cdot \sum_{t=1}^N e_t^2 \quad (6.21)$$

Ao elevar o erro ao quadrado, as parcelas da soma não se cancelarão como na métrica ME. Além disso, tende-se a *punir* mais rigorosamente erros altos do que baixos.

A métrica MSE é similar à variância, uma vez que ambas medem o desvio médio quadrático. A diferença é que o MSE mede a distância ou espalhamento entre as observações e previsões, enquanto que a variância mede a distância entre a média e as observações de uma série temporal.

Analogamente ao desvio padrão, pode-se utilizar o RMSE – *Root Mean Square Error*, dado pela Equação 6.22. Esta fórmula tem a vantagem de trabalhar na mesma unidade da observação (e.g., graus Celsius).

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{t=1}^N e_t^2} \quad (6.22)$$

### 6.5.1 Ajuste de parâmetros

Há uma miríade de condições que podem afetar uma série temporal amostrada através de um sensor no ambiente de um *datacenter*. O perfil térmico varia dependendo da topologia, configuração de servidores e carga computacional nos CPDs. Mesmo considerando um *datacenter* específico, o comportamento do perfil térmico pode variar com o tempo ou localização do sensor. Por conta desses fenômenos, é improvável que um dado conjunto de parâmetros de uma técnica de previsão possua um desempenho adequado em todas as situações de interesse.

Os métodos apresentados neste capítulo podem ser separados entre *auto-ajustáveis* ou *fixos*. A primeira categoria inclui os métodos baseados em redes neurais, que procuram minimizar o erro de suas previsões através do ajuste periódico dos pesos. Já o segundo grupo inclui os métodos Holt e AR que não possuem tal mecanismo de ajuste.

O Levenberg-Marquardt (LM) é uma heurística de minimização capaz de lidar com funções lineares e não-lineares. Neste trabalho, a técnica varia os parâmetros do método de previsão a fim de minimizar o erro (MSE) entre a temperatura prevista e observada.

O uso periódico do LM para a calibração de um método de previsão não é uma ideia nova. Um trabalho anterior [43] utilizou com sucesso o LM juntamente com o Holt para a previsão de carga computacional. Notou-se também que o gasto computacional apresentado pela técnica não é proibitivo para o seu uso em tempo-real.

O `lmfit` [39] é uma biblioteca para minimização e *fitting* de funções<sup>1</sup>. Tal biblioteca é invocada passando uma função a ser minimizada, uma lista de parâmetros (que correspondem ao ponto de partida da heurística) e uma sequência de dados que denominou-se de janela de treinamento. O objetivo é encontrar parâmetros que minimizem o erro entre a função e a janela de treinamento. Sendo assim, o `lmfit` determina o melhor conjunto

<sup>1</sup>A sua implementação é feita em Python e as funcionalidades mais intensivas computacionalmente na linguagem C. No entanto, o método LM não está restrito a linguagem Python ou C.

de parâmetros para uma técnica de previsão (Holt ou AR) sobre a janela de treinamento. Neste trabalho, a janela de treinamento é fixa e tem o mesmo tamanho que o período de aplicação do `lmfit` (de modo que dados não são utilizados mais de uma vez para treinamento). Para diferenciar os previsores fixos daqueles onde há ajuste do `lmfit`, utiliza-se os nomes `AdaptableHolt` e `AdaptableAR`.

## 6.6 Proposta

Até o momento, apresentou-se técnicas de previsão para apenas uma série temporal. No entanto, para prever emergências no CPD deve-se levar em conta múltiplas séries temporais. Relembre que na Seção 4.2 modelou-se as medidas de temperatura monitoradas como uma série temporal multivariada através da matriz  $\mathbf{X}$ . Além disso, a Seção 4.2.2 apresentou o SPIRIT que aproxima a técnica de sumarização ACP através da equação  $\mathbf{Z} = \mathbf{WX}$ . A proposta apresentada aqui consiste em simplificar a previsão do perfil térmico ao estimar a variável oculta  $z_1$  na janela de previsão, ou seja  $z'_1(t + w)$ , ao invés de prever cada série individualmente. Além do melhor desempenho, esse esquema permite que cada série temporal seja prevista levando em conta todo o perfil térmico do sistema. Essa é uma diferença grande em relação à previsão univariada, que se baseia apenas nos dados históricos de uma série temporal.

Apesar do esquema proposto agregar dados de todas as séries temporais do ambiente de testes, apenas um subconjunto delas é de interesse para a aplicação de previsão de superaquecimentos. Um superaquecimento é definido como um avanço sobre o limiar de temperatura *inlet* de um dos servidores. Sendo assim, apenas os sensores situados na entrada dos bastidores serão previstos e comparados com o limiar de temperatura estipulado.

A Figura 6.1 sumariza o processo de previsão proposto.

O processo começa com a captura de dados da rede de sensores. A árvore de coleta entrega para o nó base amostras de temperatura que são repassadas para um servidor. A função desse servidor é atualizar o vetor  $\mathbf{x}$  com medidas que são recebidas de forma assíncrona. No entanto, o vetor é submetido ao processo de previsão a cada segundo.

O valor de cada série temporal serve de entrada para o processo de sumarização (SPIRIT). Este, por sua vez, pode gerar uma ou mais variáveis ocultas, representadas por  $z_i$ , onde  $i = 1, 2, \dots, k$ . Note que apenas a primeira variável,  $z_1$ , é utilizada como entrada para o previsor. Nos experimentos realizados na Seção 7.2, uma variável oculta

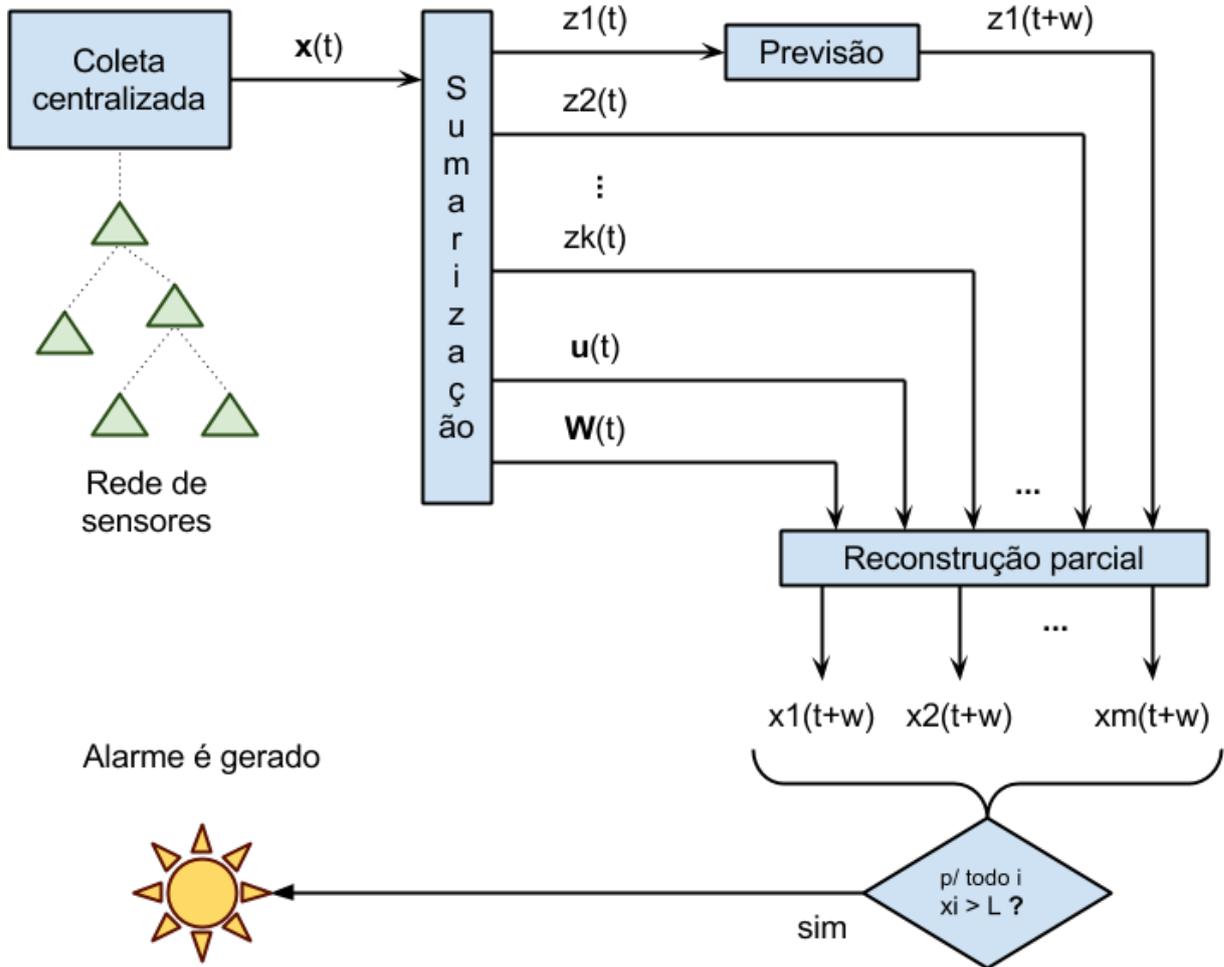


Figura 6.1: Esquema de previsão

é suficiente para manter o erro de reconstrução dentro dos limites estabelecidos. Além disso,  $z_1$  corresponde à projeção do primeiro componente principal, que representa a maior quantidade de variância das séries temporais. Portanto, as demais variáveis ocultas influenciam menos no perfil térmico projetado e podem ser passadas sem a utilização de um previsor para o processo de reconstrução. No entanto, caso o sistema apresente um número fixo de 2 ou mais variáveis, é possível adaptar o processo e instalar um módulo de previsão sobre esse subconjunto.

Tanto a média  $u(t)$  quanto os pesos  $W(t)$ , representados respectivamente como um vetor e uma matriz, também não são previstos. No caso do vetor de médias, passado tempo suficiente (e.g. 1 hora), permanece praticamente constante. A matriz de pesos não é atualizada quando chega ao processo de reconstrução. Assim como não é factível prever cada série temporal de forma individual, a previsão não é aplicada na matriz de pesos. Esse fator é uma fonte de erros, porém, para períodos de tempo curtos a correlação entre as variáveis capturadas não tende a se alterar drasticamente, viabilizando tal simplificação.

---

Ao final do processo de reconstrução, obtêm-se  $m$  variáveis que representam as séries de interesse (*inlet*). Cada valor obtido é comparado com um limiar de temperatura  $L$ , a fim de verificar se haverá um superaquecimento. Em caso positivo, um alerta seria gerado para os operadores do *datacenter*.

# Capítulo 7

## Experimentos

Neste capítulo serão apresentados os experimentos realizados no ambiente de testes para avaliar a proposta deste trabalho. O capítulo é dividido em duas seções, onde a Seção 7.1 abordará o tema de detecção de anomalias e a Seção 7.2 estudará a previsão de superaquecimentos. Os experimentos realizados em ambos os casos têm uma granularidade de 1 segundo.

### 7.1 Detecção de anomalias

A detecção de anomalias tem o objetivo de capturar mudanças significativas no perfil térmico do *datacenter*. Tais mudanças podem desencadear superaquecimentos ou apenas indicar desequilíbrios térmicos como *hotspots*. Além de identificar a anomalia, a técnica apresenta o conjunto de séries temporais associadas, permitindo assim identificar rapidamente a causa do problema (se houver).

#### 7.1.1 Ambiente de testes

*Motes* foram posicionados em vários pontos do Ambiente de Testes<sup>1</sup>. Cada *mote* possui um número de identificação como consta na Tabela 7.1, que será usado para referenciar cada aparelho, sensor e série temporal associada neste capítulo de acordo com o contexto.

Note que cada bastidor possui a mesma quantidade de *motes*, porém há uma diferença no posicionamento horizontal dos sensores. Isso se deve, meramente, a uma dificuldade de fixar os sensores na mesma posição. O Bastidor 1 possui uma porta com diversos furos (uma espécie de grade), e uma região sólida no meio. Sendo assim, a única maneira de

---

<sup>1</sup>Para uma descrição mais detalhada do espaço físico e configuração do experimento, veja a Seção 3.6.

Tabela 7.1: Posicionamento dos *motes*

<b>Bastidor 1, frente</b>	<b>Meio</b>	<b>Bastidor 1, trás</b>	<b>Esquerda</b>	<b>Direita</b>
<b>Topo</b>	249	<b>Topo</b>	248	222
<b>Meio</b>	253	<b>Meio</b>	223	227
<b>Fundo</b>	245	<b>Fundo</b>	252	233

<b>Bastidor 2, frente</b>	<b>Esquerda</b>	<b>Bastidor 2, trás</b>	<b>Esquerda</b>	<b>Direita</b>
<b>Topo</b>	241	<b>Topo</b>	216	238
<b>Meio</b>	237	<b>Meio</b>	226	214
<b>Fundo</b>	239	<b>Fundo</b>	231	217

<b>Outros</b>	<b>Identificador</b>
<b>Parede oposta aos aparelhos de ar-condicionado</b>	235
<b>Porta da sala</b>	215
<b>Teto do Bastidor 1</b>	251
<b>Fixado a saída de ar do Ar-condicionado 1</b>	228
<b>Parede, debaixo do Ar-condicionado 2</b>	242
<b>Alinhado ao Ar-condicionado 3</b>	225

fixar os *motes* com segurança era através dessa região. Já o Bastidor 2 mantinha a sua porta sempre aberta, pois possuía um terminal de controle embutido. Além disso, a região do meio deste bastidor não poderia ser utilizada para não obstruir a entrada de ar dos servidores. Portanto, os *motes* foram fixados na parede interna esquerda desse bastidor.

Uma vez que os aparelhos de ar-condicionado injetam ar frio e retiram ar quente pelo topo da sala, foram posicionados *motes* para capturar a temperatura de entrada e saída. Para o Ar-condicionado 1 (ou Ar-1), foi posicionado um *mote* no teto do Bastidor 1, alinhado com a saída de ar. Além disso, posicionou-se o *mote* 228 de forma adjacente à saída de ar. Dessa forma, é possível medir a diferença de temperatura entre o ar que é injetado com aquele que chega no bastidor. O *mote* 242 foi colocado na parede, abaixo do segundo ar-condicionado. Já o *mote* 225 está posicionado sobre uma caixa, a direita do Bastidor 2, alinhado com a saída de ar do Ar-3.

O *mote* 235 foi posicionado na parede oposta dos aparelhos de ar-condicionado com a finalidade de medir a diferença de temperatura entre o ar que reflete na parede e aquele que entra nos bastidores. Já o *mote* 215 tem dois propósitos: medir a temperatura fora da sala (que pode ser considerada a temperatura ambiente) e detectar mudanças de temperatura que ocorrem devido à abertura da porta da sala.

### 7.1.2 Experimentos

A fim de testar a técnica proposta montaram-se experimentos onde eventos térmicos foram propositalmente causados. Tais eventos foram planejados para causar uma mudança significativa no perfil térmico do *datacenter* sem causar um superaquecimento.

No Evento A, a porta da sala do ambiente de testes é deixada totalmente aberta por 16 minutos. Esse evento simula o caso onde há manutenção no CPD ou quando não há um mecanismo de fechamento automático da porta e esta pode ficar aberta por descuido. Tal evento gera uma ineficiência térmica pois o ar de fora da sala se mistura com o ar frio do ambiente de testes.

Para o Evento B, deixa-se o Ar-condicionado 1 desligado por 6 minutos. Tal evento é mais agressivo que o anterior, pois foi simulada uma falha temporária no aparelho de arrefecimento que está associada ao fenômeno de ilha de calor.

Foram realizados dois experimentos com condições similares, onde o Evento A precede o Evento B. No entanto, a carga de trabalho no ambiente de testes não é controlada. Por conta disso, variações nos experimentos podem ocorrer. Através da Tabela 7.2, pode-se consultar os intervalos de duração de cada evento.

Tabela 7.2: Intervalos dos eventos térmicos

<b>Evento</b>	<b>Intervalo</b>
Experimento I: Evento A	15:11 – 15:27
Experimento I: Evento B	16:24 – 16:30
Experimento II: Evento A	15:45 – 16:01
Experimento II: Evento B	17:16 – 17:22

#### 7.1.2.1 Experimento I

A Figura 7.1 mostra o *trace* levantado durante o Experimento I. A maioria das séries apresenta pouca variação de temperatura, destacando-se os *notes*: 252 (azul), 227 (verde) e 228 (vermelho). As duas primeiras estão associadas à áreas na parte de trás do Bastidor 1 que sofrem influencia de uma carga de trabalho variável. Já a última mede a saída de ar do Ar-1, que sofre variações de temperatura provenientes do ar quente dos servidores.

A Figura 7.2 mostra as anomalias encontradas e os eventos provocados. Note que não se deve avaliar a qualidade das previsões através da sobreposição entre eventos e anomalias, por conta da inércia térmica e retorno ao equilíbrio térmico.

Tabela 7.3: Anomalias do Experimento I

<b>Anomalia</b>	<b>Motes associados e pesos</b>
A1 – 10 min após abertura da porta.	228 (0,79), 227 (0,39), 223 (0,31)
A2 – 4 min após religamento do Ar-1.	252 (0,87), 227 (0,34)
A3 – 10 min após religamento do Ar-1.	228 (0,73), 252 (0,61)

As anomalias encontradas no Experimento I foram rotuladas na primeira coluna da Tabela 7.3, em ordem cronológica. Na segunda coluna dessa tabela, tem-se os *motes* que estão associados à cada anomalia. Em todos os casos em que houve anomalias no experimento foi necessário apenas uma variável-oculta extra para *explicar* o fenômeno. Entre parênteses, é exibido o valor absoluto dos pesos em que cada série apresenta em sua segunda variável oculta (ver Seção 4.2.2 para mais detalhes). Para diminuir o escopo desta análise, apenas os pesos com magnitude maior que 30% foram considerados.

Observe que A1 ocorre 10 minutos após a abertura da porta, em um momento em que há um pico de temperatura no *mote* 252 e um vale no *mote* 228. Essa mudança de correlação é o que ativa a segunda variável oculta, que tem o *mote* 228 (Ar-1) como principal responsável. Outros dois *motes* da área traseira do Bastidor 1 (223 e 2273) também estão presentes na anomalia. Essa anomalia é efeito da mistura de ar de fora da sala com o ar de dentro.

O desligamento do Ar-1 é seguido de duas anomalias, cada uma com uma causa distinta. No caso de A2, trata-se de uma queda repentina de temperatura na série 252 que contrasta com a descida suave da maioria das outras séries temporais. A3 é influenciada pelos *motes* 228 e 252 em uma escala similar. Enquanto o primeiro apresenta um vale repentino, o segundo apresenta uma subida elevada de temperatura. De modo geral, o comportamento de ambas anomalias é similar a A1, que está associado à saída de ar e a sensores localizados na parte de trás do Bastidor 1.

### 7.1.2.2 Experimento II

A Figura 7.3 apresenta um comportamento mais estável em comparação com o Experimento I. Além de estar sujeito a uma carga de trabalho diferente daquela observada no experimento anterior, o gráfico se apresenta em uma escala de tempo menor, contribuindo para uma aparência mais estável.

As anomalias encontradas neste experimento estão identificadas na Tabela 7.4. A anomalia A4 ocorre logo após o início do Evento A e é causada principalmente por sensores

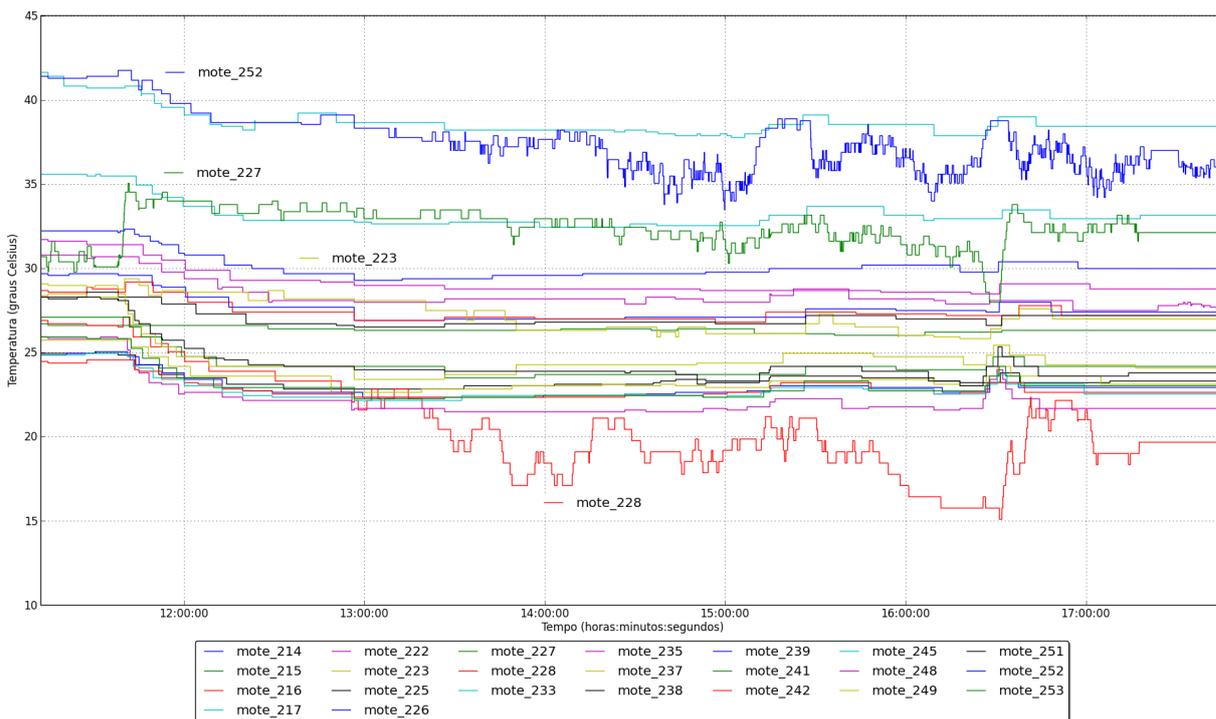


Figura 7.1: Experimento I – *trace*

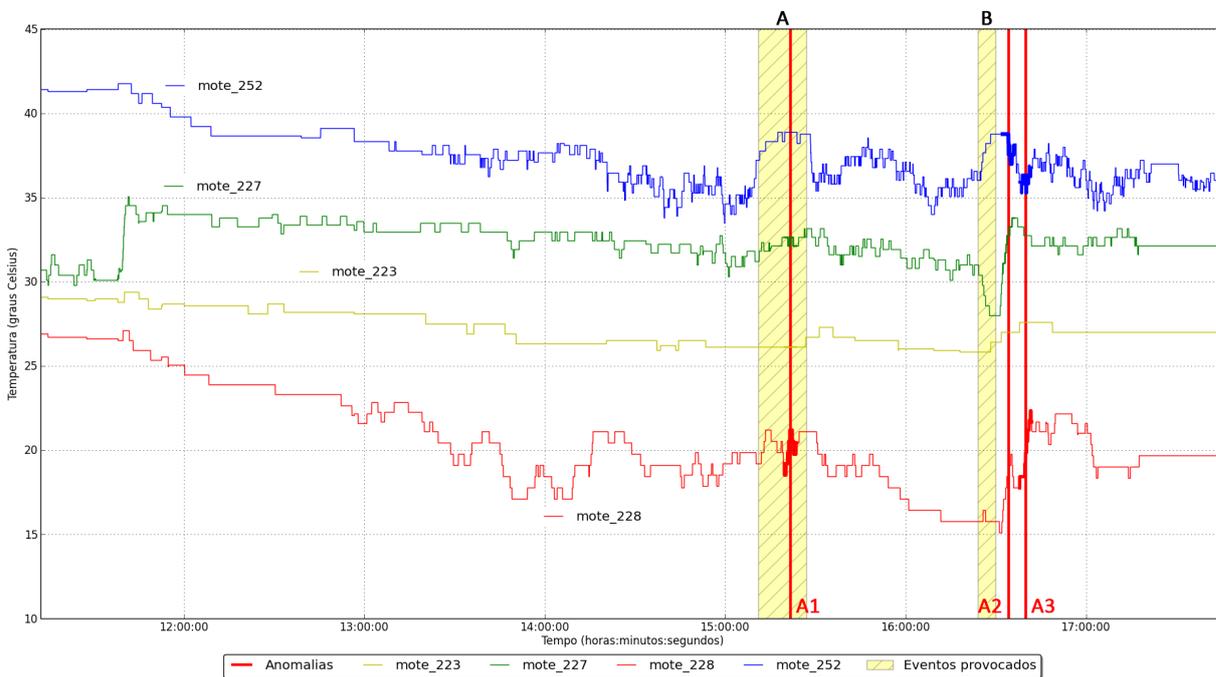


Figura 7.2: Experimento I – anomalias esperadas e detectadas

na parte traseira do Bastidor 1 (223 e 248) e Bastidor 2 (238). É possível observar que o gráfico das séries 223 e 248 é praticamente constante até o momento onde a anomalia foi detectada.

A segunda anomalia, A5, aparece no momento em que a temperatura começa a di-

Tabela 7.4: Anomalias do Experimento II

<b>Anomalia</b>	<b>Motes associados e pesos</b>
A4 – 1 min após abertura da porta	223 (0,69), 238 (0,41), 248 (0,31), 235 (0,30)
A5 – 6 min após fechamento da porta	251 (0,70), 228 (0,68)
A6 – 2 min após religamento do Ar-1	228 (0,83)

minuir e se estabilizar por conta do fechamento da porta. Ambos os *motes* identificados estão relacionados ao Ar-1: o mote 251, localizado no topo do Bastidor 1 e o mote 228 na saída de Ar-1.

A anomalia A6 ocorre 2 minutos após o religamento do Ar-1 e ocasiona um aumento intenso de temperatura no mote 228. (Um efeito parecido também pode ser observado no Experimento I.) A natureza deste efeito é contra-intuitiva, pois ar-frio está sendo injetado na sala por conta do religamento do ar. Em um experimento não mostrado aqui, testou-se outro *mote* na mesma posição de 228 e notou-se um efeito similar, indicando que não se trata de um defeito.

Uma hipótese<sup>2</sup> para explicar esse fenômeno se baseia no fluxo de ar presente na sala. No início do Evento A, nota-se uma diminuição de temperatura do *mote* 228 no Experimento II. Os aparelhos de ar instalados na sala criam uma corrente de ar que suga ar quente. Por conta da sua localização, o Ar-1 afeta principalmente o ar da parte traseira do Bastidor 1. Uma vez que ocorre seu desligamento, a corrente de ar predominante passa a ser dos aparelhos 1 e 2. Sendo assim, o *mote* 228 detecta uma queda de temperatura pois o ar quente do Bastidor 1 está sendo desviado para outra região. De maneira simétrica, há um aumento de temperatura ao religar o Ar-1 pois o mesmo desloca ar quente mais rápido do que consegue refrigerar aquela região.

Esse experimento teve um comportamento similar ao anterior, onde a grande maioria dos sensores associados aos eventos estava localizada na parte de trás do Bastidor 1 ou relacionada ao Ar-1.

### 7.1.3 Conclusão

Uma parte importante desta análise é apontar as séries temporais responsáveis pelas anomalias encontradas. Para algumas dezenas de *motes*, a inspeção manual para determinação da causa da anomalia é suficiente. No entanto, a medida que mais séries estão

<sup>2</sup>Note que essa hipótese não é suportada por experimentos. Trata-se apenas de um cenário plausível para contabilizar esse efeito.

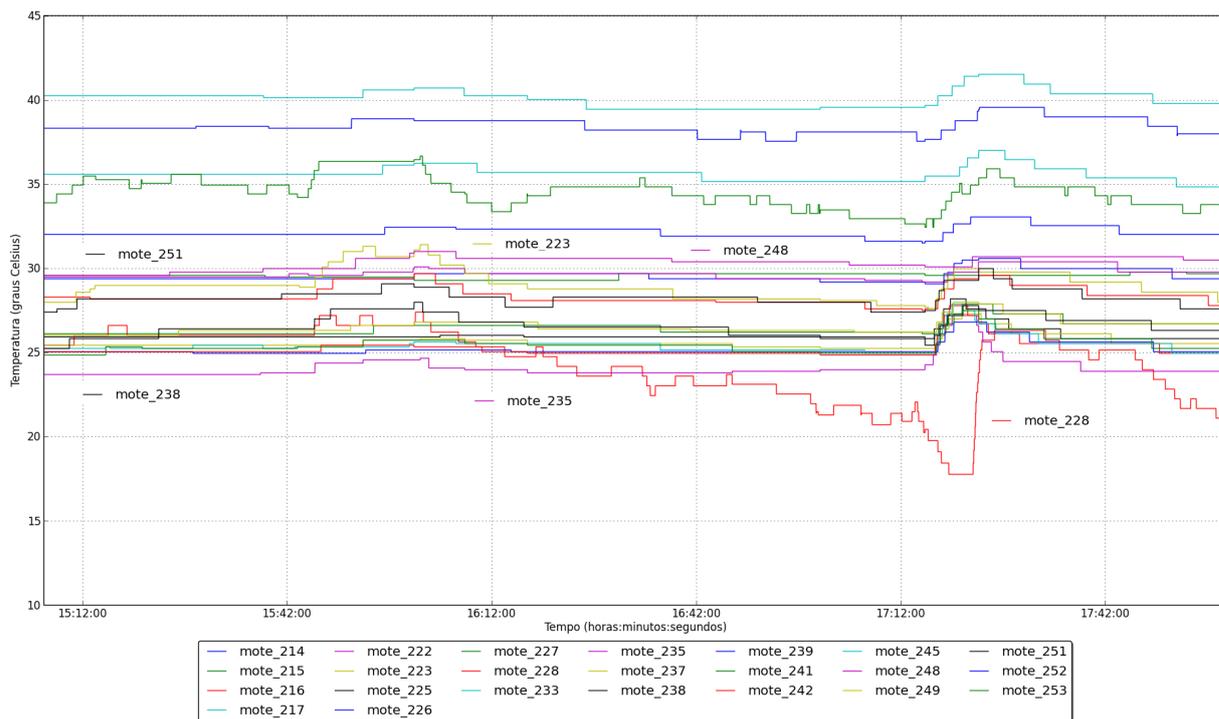


Figura 7.3: Experimento II – trace

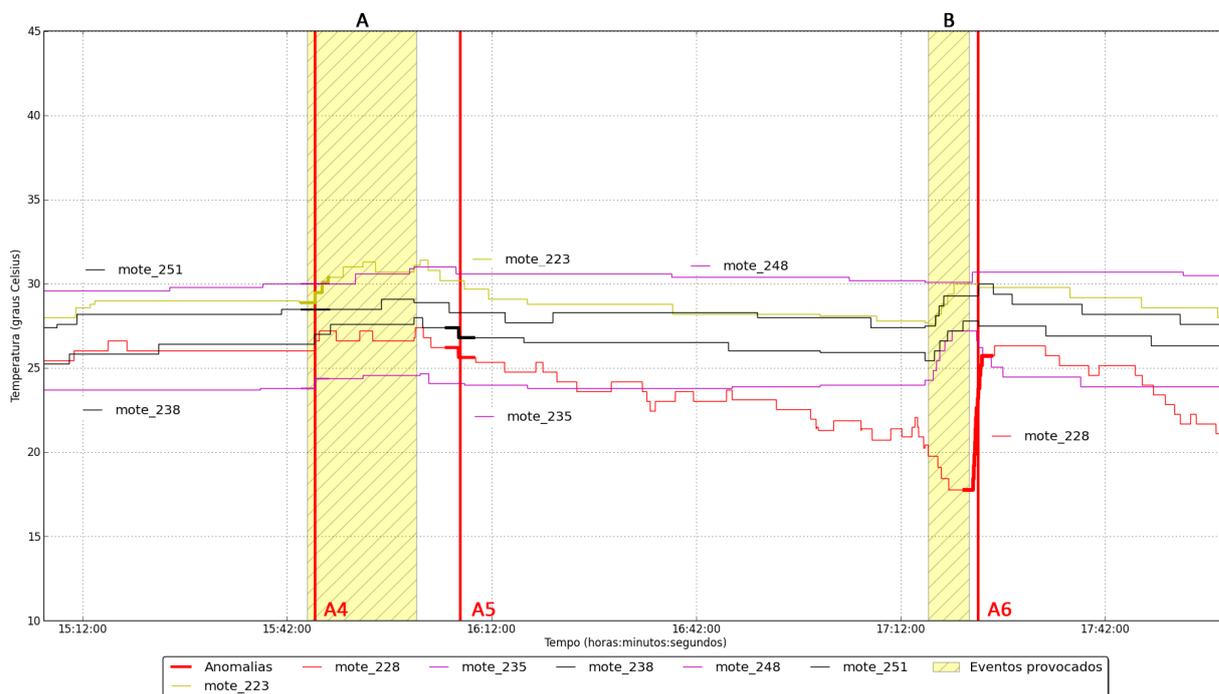


Figura 7.4: Experimento II – anomalias e eventos

presentes no conjunto de dados, é infactível realizar uma análise manual.

As anomalias identificadas (A1–A6) foram associadas aos Eventos A e B em todos os experimentos, não havendo nenhum falso-positivo. No entanto, houve casos onde as anomalias foram identificadas apenas após o fim de um evento, como no caso do Expe-

rimento I. Esse aspecto pode ser um reflexo da sensibilidade dos parâmetros do SPIRIT. Enquanto é possível aumentar a sensibilidade do SPIRIT para incrementar o número de detecções nesses *traces*, tal ação tende a gerar falsos-positivos. O ajuste fino desses parâmetros depende do ambiente e do nível de alertas esperado pelos responsáveis pelo *datacenter*. Sendo assim, esta análise fica além do escopo deste trabalho. Outra possibilidade é que, devido a curta duração dos Eventos A e B há uma inércia térmica até o momento em que o evento impacte o perfil térmico.

Um caso particular é o início do Evento B, onde não houve a detecção em ambos os experimentos. É possível que esse efeito seja causado por conta dos parâmetros utilizados, duração do evento (que é mais curto que o Evento A) ou simplesmente acaso. É possível também que o Evento B tenha um impacto mais homogêneo (não havendo a quebra de correlação entre as séries, não há detecção de anomalias) entre os sensores do que no Evento A, onde os sensores próximos a porta são mais diretamente afetados.

Ao detectar os *motes* associados a cada anomalia, é possível reduzir consideravelmente o número de séries a serem investigadas. Através dessa redução é possível facilitar a interpretação e conseqüentemente a resposta à uma anomalia.

Outra vantagem da técnica proposta é a capacidade de identificar a região afetada pela anomalia. Na grande maioria dos casos analisados (A1–A6) os *motes* associados estavam na parte de trás dos bastidores ou associados de alguma forma ao Ar-1. No entanto, não é possível identificar diretamente o tipo de anomalia ocorrida apenas pelo conjunto de sensores afetados. De qualquer forma, seria necessário repetir os experimentos mais vezes e talvez com mais sensores para obter uma resposta mais definitiva sobre isso.

## 7.2 Previsão de superaquecimentos

Realiza-se a previsão de temperatura no CPD com o objetivo de antecipar e combater superaquecimentos. A fim de testar as técnicas apresentadas no Capítulo 6, coletou-se *traces* de temperatura do ambiente de testes (ver Seção 6). Tais *traces* servirão de entrada para um simulador que permitirá examinar diversas métricas de interesse para a avaliação deste trabalho.

De modo geral, superaquecimentos não podem ser provocados por pesquisadores dado o risco de danificar o equipamento. Sendo assim, trabalhos anteriores limitaram-se a utilizar simulações de temperatura [36] ou *traces* normais de operação onde supõe-se um estado de superaquecimento [25]. No entanto, este trabalho capturou um estado de

superaquecimento provocado por uma falta de energia no ambiente de testes.

### 7.2.1 Ambiente de testes

A Figura 7.5 mostra a temperatura de vários sensores durante um intervalo em que ocorreu falta de energia no ambiente de testes (ver Seção 3.6). Note que a temperatura começa a subir acentuadamente por volta de 06:30, onde ocorre a falta de energia. Os servidores permanecem ligados – e dissipando calor – pois estão conectados a um sistema UPS (*Uninterruptible Power Supply*). Porém, os três aparelhos de ar condicionado não estão conectados ao sistema de energia reserva e foram desligados imediatamente.

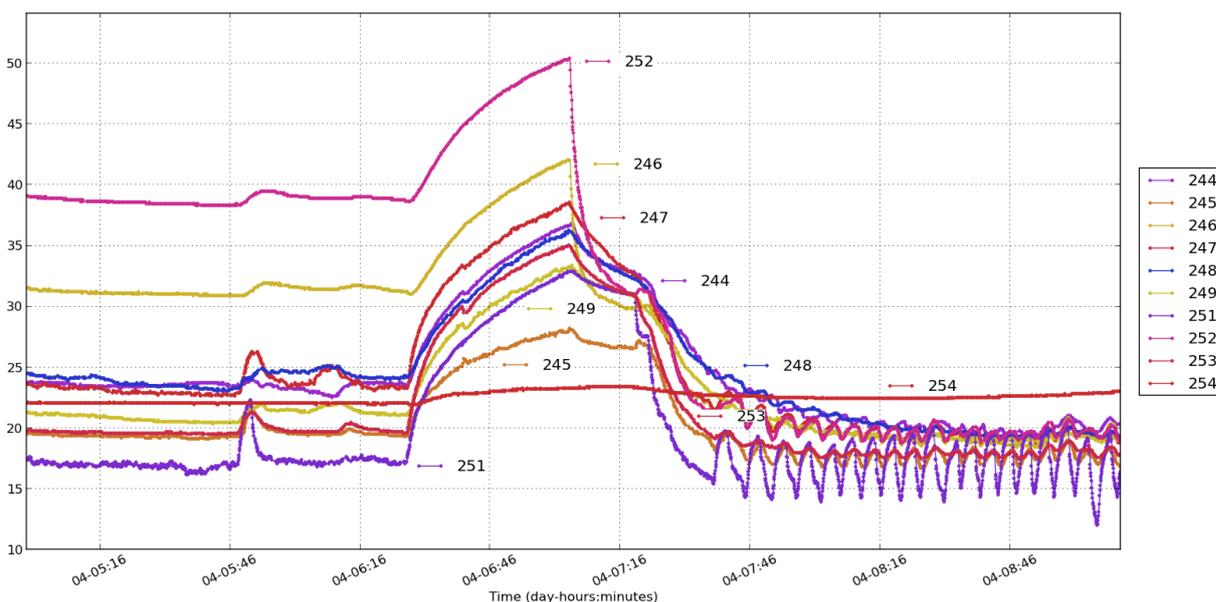


Figura 7.5: *Trace* de temperatura durante a falta de energia

Em seguida, por volta de 07:00, algumas máquinas começam a ser desligadas. Às 07:30 não há nenhuma máquina ligada. Máquinas que cruzem um limiar térmico configurável do processador (60 °C) ou placa-mãe (40 °C) são desligadas automaticamente. Além disso, servidores podem ser desativados caso a energia do sistema UPS se esgote. Caso o sistema proposto estivesse implementado durante esse superaquecimento, um operador poderia eventualmente ser avisado com antecedência e tomar uma atitude. Tal ação pode evitar que dados importantes sejam perdidos ao salvar o processamento feito em disco e desligar as máquinas remotamente.

Note que a técnica de compressão de dados da Seção 3.4 não está ativa durante o *trace*. No entanto, emulou-se tal política via software de modo a avaliar o desempenho das técnicas de previsão nessas condições.

A previsão é feita sobre o *output* do SPIRIT ao invés da temperatura de cada série temporal individual. Além disso, somente os *motes* da entrada de ar (245, 249 e 253) foram usados. Consulte a Seção 6.6 para mais detalhes.

O ambiente de testes desta seção possui uma configuração de *motes* distinta daquela apresentada na Seção 7.1. Este experimento conta com 10 sensores que estão descritos na Tabela 7.5.

Tabela 7.5: Localização dos *motes* no ambiente de testes

<b>Localização</b>	<b><i>Motes</i></b>
<i>Inlet</i> do Bastidor 1	245, 249, 253
Saída do Bastidor 1	252, 246, 248, 247
Saída do Bastidor 2	244
Sobre o Bastidor 1	251
Alinhado à saída do Ar-2	254

## 7.2.2 Ajuste de parâmetros dos modelos

Nesta seção serão realizados experimentos para determinar um tamanho adequado para a janela de treinamento com o método LM, discutido na Seção 6.5.1. O tamanho dessa janela deve ser grande o suficiente para não ocasionar *overfitting*, de modo que os parâmetros obtidos generalizem o treinamento fornecido. Em contrapartida, é necessário escolher uma quantidade de dados que seja pequena o suficiente para responder às mudanças de comportamento que ocorrem nas séries temporais em caso de superaquecimentos. Sendo assim, é necessário investigar um tamanho adequado para cada janela de treinamento (AR e Holt).

Para o esquema proposto na Seção 6.6, um RMSE é calculado para cada série temporal reconstruída correspondente ao *inlet*. De modo a obter uma estatística que represente o erro do sistema, realiza-se a média entre os RMSEs de cada série.

Para investigar o erro da previsão com o LM, realizou-se um experimento onde variou-se as janelas de treinamento e previsão. A variação foi de 1 a 10 minutos e 1 a 12 minutos com incrementos de 1 minuto, para as janelas de previsão e treinamento, respectivamente. As Figuras 7.6 e 7.7 correspondem ao resultados para AdaptableAR e AdaptableHolt, respectivamente.

A Figura 7.6 mostra uma ligeira diferença entre as janelas de treinamento medidas no AdaptableAR. A curva que apresenta menor erro médio ocorre para a janela de 10 minutos.

Já a Figura 7.7, referente ao `AdaptableHolt`, possui um comportamento similar porém com um erro médio menor que `AdaptableAR`. O seu valor ótimo ocorre em 8 minutos. Ambas as figuras apresentam um comportamento parecido em relação a janela de previsão: quanto mais distante for a previsão realizada, maior o erro.

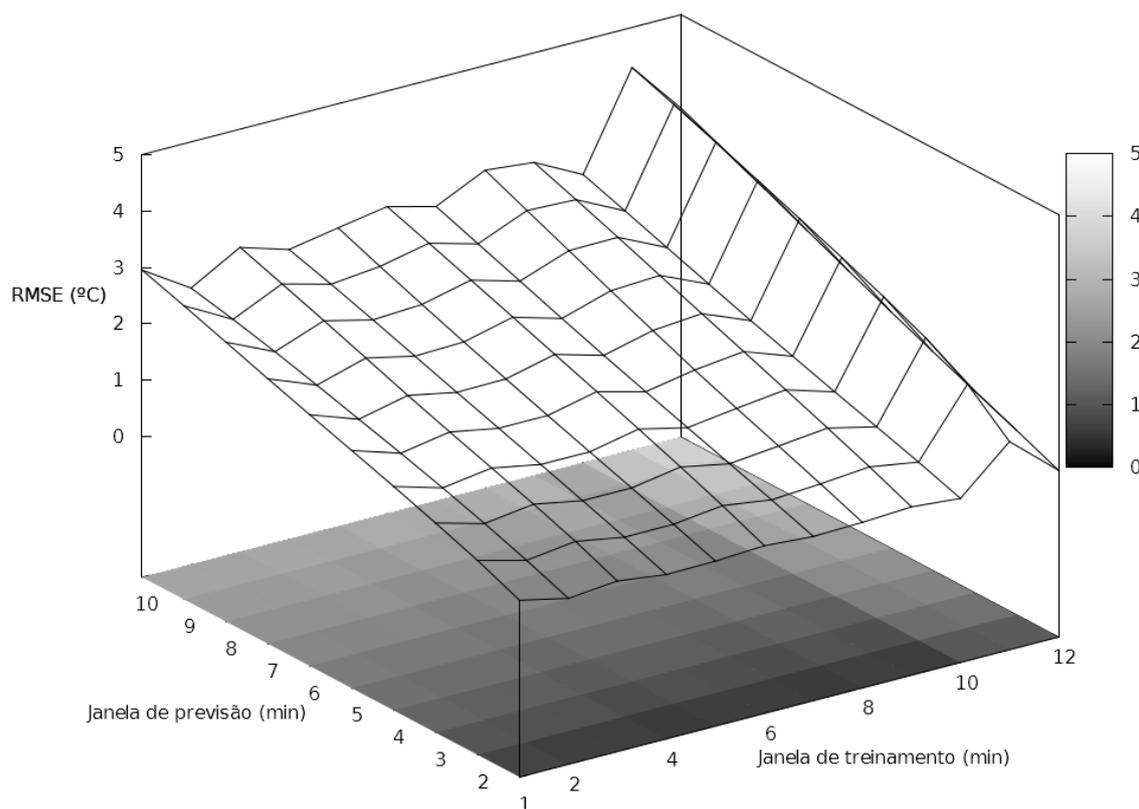


Figura 7.6: Erros em previsões – `AdaptableAR`

Para as redes neurais, da categoria de previsores auto-ajustáveis, estudou-se o parâmetro *taxa de aprendizado*. A magnitude dessa variável controla a adaptação da rede ao erro causado pelas previsões. Quando seu valor se aproxima de 1, os pesos da rede neural se adaptam rapidamente de modo a gerar a previsão correta. No entanto, corre-se o risco de *overfitting*. Já valores próximos de 0 se aproximam-se do previsor Naïve.

Observando a Figura 7.8, referente ao `LinearPerceptron`, nota-se um comportamento bastante uniforme. As taxas de aprendizado maiores que  $10^{-3}$  não foram mostradas pois houve uma degeneração do perceptron e erros muito altos. Um padrão que o gráfico revela é um baixo erro para uma taxa de aprendizado de  $10^{-3}$ , um valor fixo para os valores intermediários e, finalmente, um crescimento no erro a partir de  $10^{-10}$ .

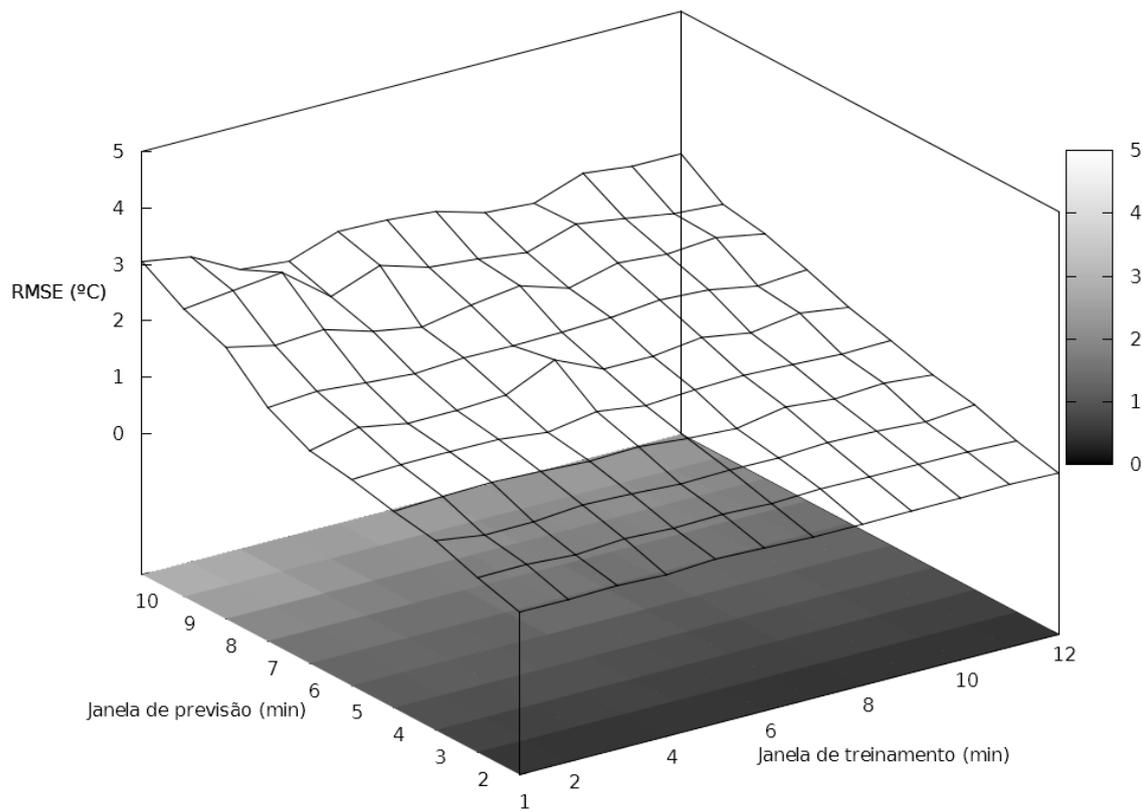


Figura 7.7: Erros em previsões – AdaptableHolt

A Figura 7.9, associada ao RollingPerceptron, apresenta um comportamento parecido com a anterior. No entanto, observa-se um erro alto para um aprendizado de  $10^{-3}$ . Apesar da diferença entre os erros a partir deste ponto ser pouca, identificou-se que, em média, o erro é menor para uma taxa de aprendizado de  $10^{-5}$ . Dado que o RollingPerceptron requer a ativação da rede várias vezes para gerar uma previsão, não é surpresa que a sua taxa de aprendizado ideal seja menor do que LinearPerceptron.

### 7.2.3 Acurácia de previsão dos modelos

A análise da seção anterior permitiu que fosse possível calibrar o tamanho da janela de treinamento ou taxa de aprendizado ideal para cada método. Estimou-se os valores de 10 e 8 minutos para a janela de treinamento dos previsores AdaptableAR e AdaptableHolt, respectivamente. Para os perceptrons, estimou-se  $10^{-3}$  e  $10^{-5}$  para a taxa de aprendizado de LinearPerceptron e RollingPerceptron, respectivamente. Tais valores serão utilizados nos experimentos do restante do capítulo.

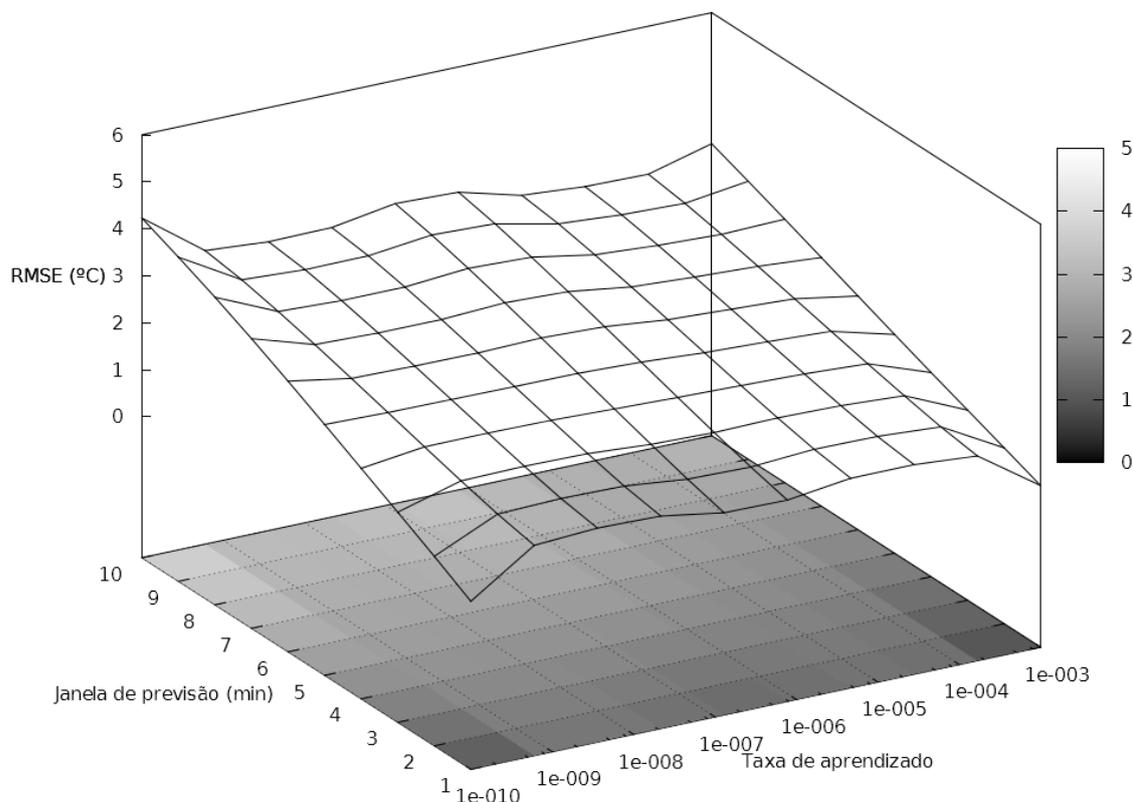


Figura 7.8: Erros em previsões – LinearPerceptron

Dado que os previsores foram calibrados, podemos avaliar seu desempenho. Uma preocupação fundamental é mensurar a acurácia com que os previsores modelam o processo térmico do *datacenter*. Isso será feito comparando o RMSE resultante das diversas abordagens.

A Figura 7.10 sumariza os resultados obtidos para várias janelas de previsão. Note que o erro aumenta de maneira conjunta com a janela de previsão, um comportamento esperado devido a dificuldade crescente de se realizar previsões de longo prazo aliada a ausência de eventos periódicos.

A maioria dos previsores estudados obtém um RMSE menor do que o método Naïve, com a exceção do LinearPerceptron. Os ganhos se dão com mais evidência a medida que a janela de previsão aumenta. Os previsores com os modelos mais precisos são AdaptableHolt e RollingPerceptron, respectivamente.

O RMSE mede a dispersão das previsões em torno das observações de temperatura. Para uma janela de previsão suficientemente grande, o RSME se tornará tão alto que a

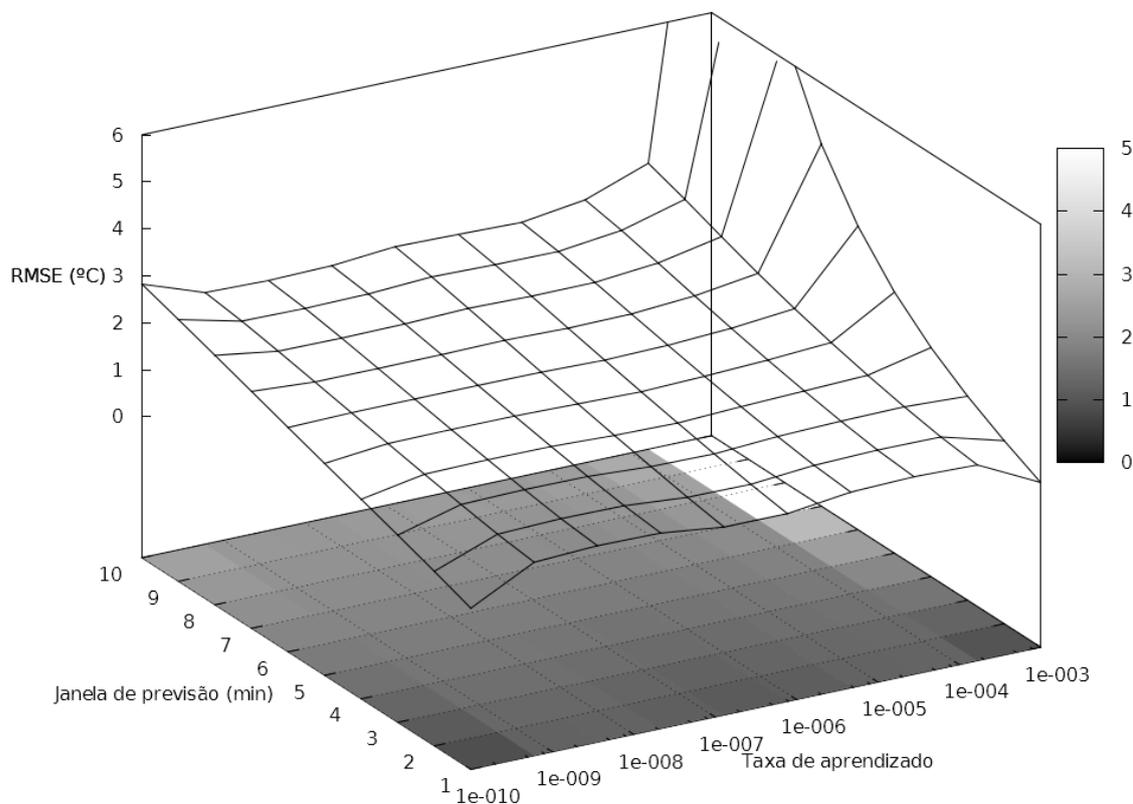


Figura 7.9: Erros em previsões – RollingPerceptron

utilidade do previsor será nula. Sendo assim, é necessário limitar a janela de previsão utilizada de acordo com um valor de RMSE máximo aceitável. Trata-se de um valor arbitrário que está relacionado com a confiabilidade que um previsor apresenta. Neste trabalho, utilizou-se um limite de 2 °C. Sendo assim, devemos utilizar uma janela de previsão de no máximo 6 ou 9 minutos (aproximadamente), dependendo do previsor.

#### 7.2.4 Análise do tempo de antecedência

O modelo de previsões de superaquecimentos exposto neste trabalho requer um passo extra além da previsão de temperatura. Deseja-se analisar a antecedência estimada para um *redlining* iminente. Através desse tempo de antecedência, será possível planejar ações para evitar a desestabilização do serviço do *datacenter*.

O Tempo de Antecedência (TA) será definido como o maior intervalo entre um alarme e o início de um superaquecimento real. O alarme, por sua vez, é simplesmente um instante no futuro onde o previsor estima que haverá uma temperatura maior que o limiar pré-

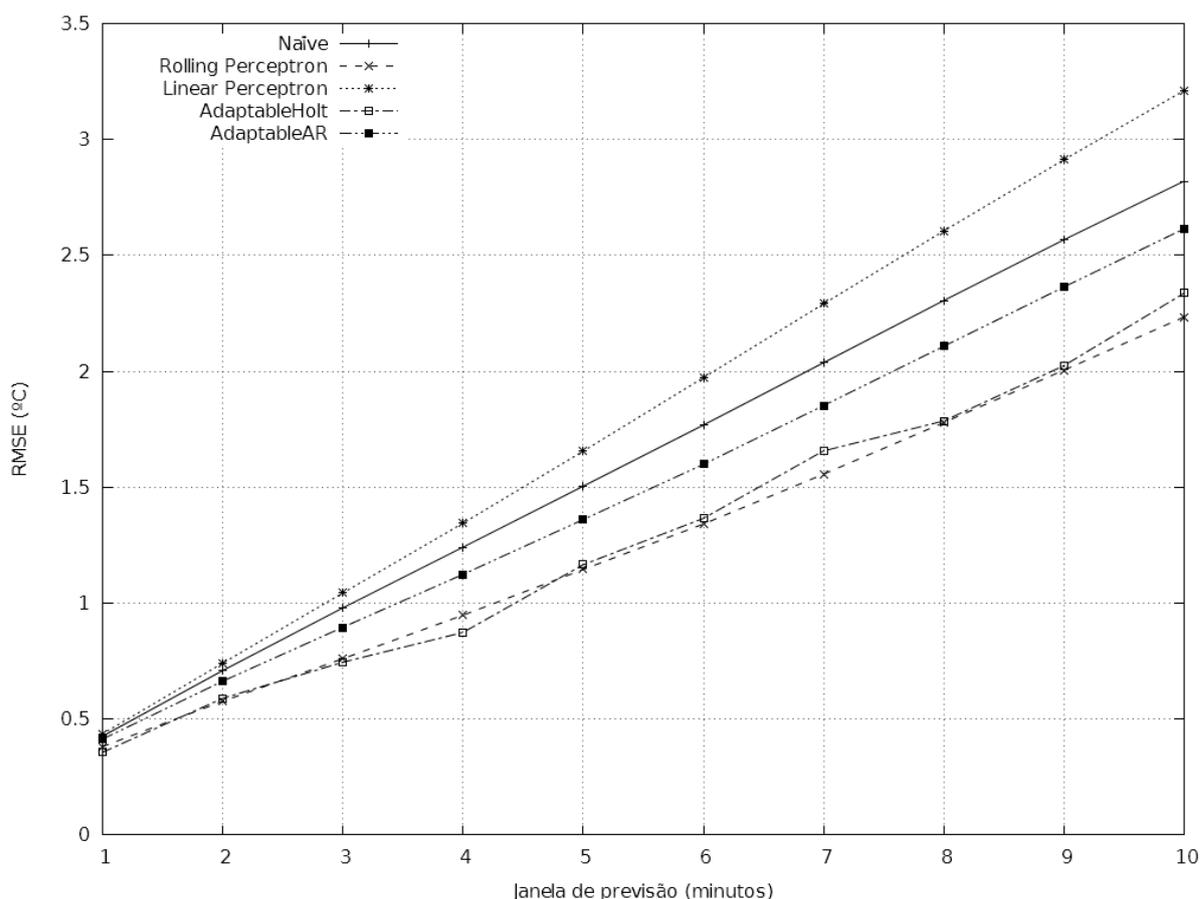


Figura 7.10: Erros de temperatura em várias janelas de previsão

estabelecido de temperatura,  $L$  (que vale  $30\text{ }^{\circ}\text{C}$ ). Similarmente, o superaquecimento é definido quando a temperatura de fato ultrapassa  $L$ .

Como o esquema proposto trata de várias séries, a verificação da ultrapassagem do limiar deve ser realizada em todas as séries de interesse (*inlet*) após reconstrução pelo SPIRIT. Basta que uma delas ultrapasse  $L$  para configurar um alarme.

Outra métrica importante a ser analisada é a quantidade de tempo em que alarmes foram emitidos, porém sem haver nenhum superaquecimento naquele instante. Esta métrica reflete a confiabilidade que está associada a cada alarme. Nomeia-se esta métrica de Falsos-Positivos (FP) e é definida pela soma de todos os alarmes onde superaquecimentos não são encontrados.

Uma vez que é improvável acertar o momento exato do início de um superaquecimento, é importante criar uma região de tolerância ao redor deste instante. Projetou-se um limiar que é igual a 25% do tamanho da janela de previsão. Por exemplo, um superaquecimento que ocorre no tempo  $t_e$  não terá alarmes contabilizados como FP, para uma janela de previsão de 8 min, na região  $[t_e - 2, t_e + 2]$  min.

Para esclarecer a interação entre as métricas FP e TA, considere os seguintes exemplos. Seja um previsor que, para todo instante de tempo, faça a previsão de temperatura  $p_{t+w} = L + 1$ . Nesse caso, o tempo de antecedência é  $w$ , uma vez que os instantes de superaquecimento foram antecipados. No entanto, note que em todos os instantes o previsor gera alarmes, criando uma quantidade enorme de falsos positivos. Sendo assim, pode-se rejeitar tal previsor baseado no resultado de FP. Simetricamente, considere um previsor que gere previsões onde  $p_{t+w} = L - 1$ . Agora o número de falsos positivos é zero, porém TA também é zero. Isso significa que o previsor é inútil, já que não é capaz de anteceder superaquecimentos, e deve ser descartado.

Analisando os experimentos, exceto no caso do LinearPerceptron, pôde-se observar um TA equivalente e próximo ao limite em todos os casos. A diferença, portanto, se resume a taxa de FP que cada previsor apresenta. Curiosamente, AdaptableAR possui o menor valor de FP de todos os métodos, apesar ter apresentado anteriormente um RMSE superior a AdaptableHolt e RollingPerceptron. AdaptableHolt, por sua vez, apresentou os maiores índices de alarmes falsos, apesar de variar bastante o seu valor dependendo da janela de previsão.

### 7.2.5 Conclusão

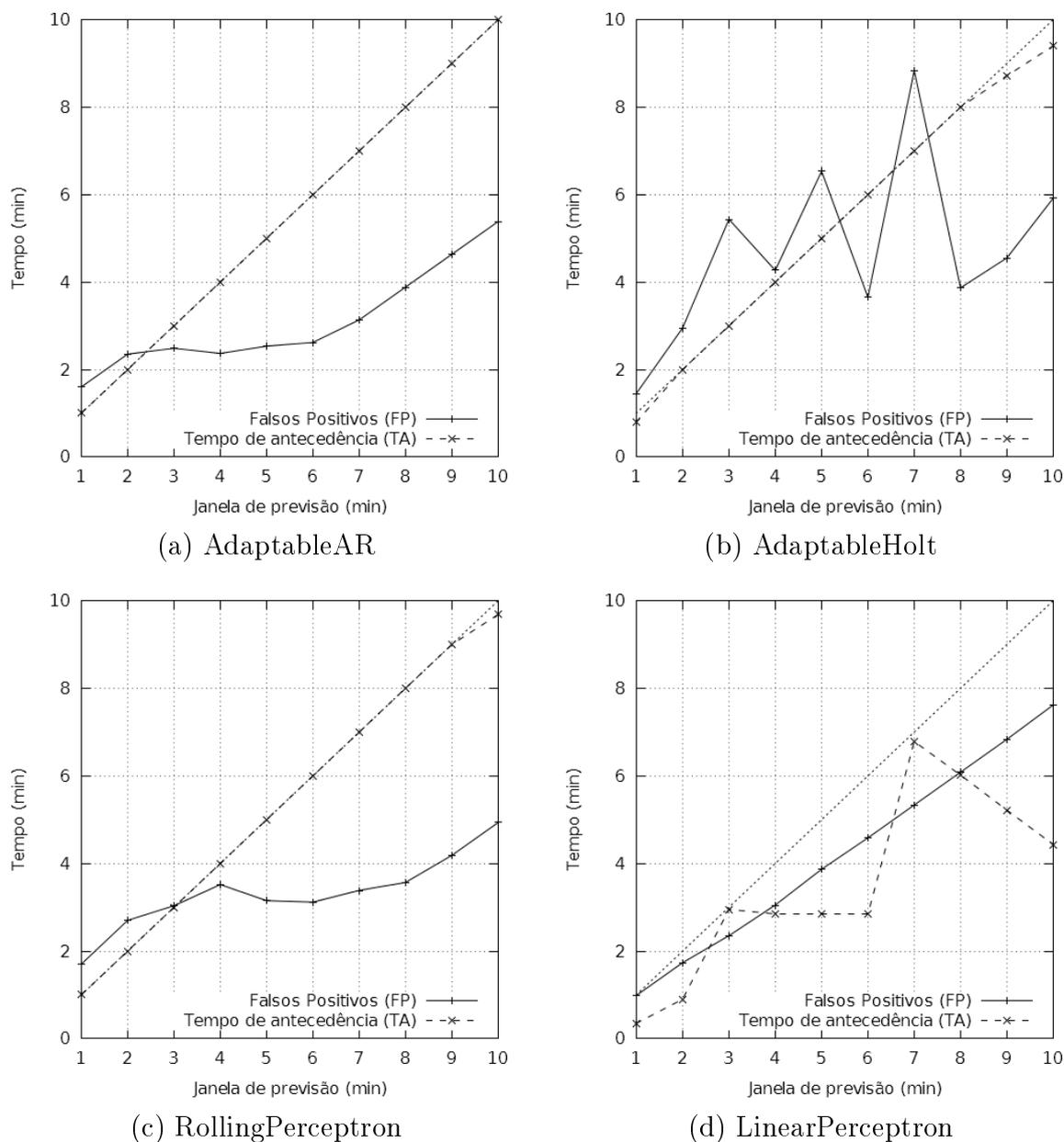
O estudo realizado neste capítulo comparou diversos previsores sobre diferentes aspectos. A análise de acurácia dos previsores revelou que a janela de previsão máxima é de 6 min para o AdaptableAR e 9 min para AdaptableHolt e RollingPerceptron de forma a respeitar o limiar de 2 °C. Já o estudo do tempo de antecedência revelou que AdaptableAR apresenta os menores índices de falsos alarmes, enquanto os demais previsores encontram-se em uma situação equivalente no quesito tempo de antecedência.

A Tabela 7.6 sumariza os resultados obtidos até então. Uma vez que AdaptableAR ultrapassa o limiar de 2 °C em um instante entre 7 e 8 min, calculou-se os resultados com uma janela de previsão de 7,5 min. Em todos os casos, buscou-se respeitar o limiar estabelecido de RMSE máximo, que vale 2 °C.

Apesar de apresentar a menor quantidade de FP, AdaptableAR tem um valor de TA inferior aos outros métodos. Por sua vez, AdaptableHolt apresenta resultados inferiores e, sendo assim, RollingPerceptron é o candidato que tem melhor desempenho de forma qualitativa.

Sob o aspecto de implementação, o RollingPerceptron tem a vantagem por ser mais

Figura 7.11: Tempo de antecedência e alarmes falsos para vários previsores



Técnica	RMSE de previsão ( $^{\circ}\text{C}$ )	FP (s)	TA (s)
RollingPerceptron ( $w = 9$ min)	2.0048	251	540
AdaptableHolt ( $w = 9$ min)	2.0244	273	523
AdaptableAR ( $w = 7,5$ min)	1.9802	211	450

Tabela 7.6: Sumário dos resultados da previsão térmica

simples. O predictor AdaptableHolt depende da biblioteca `lmfit`, que, por sua vez, pode não estar disponível em todas as plataformas de programação. Neste caso, o programador terá um custo maior para implementar as suas funcionalidades de maneira eficiente.

---

RollingPerceptron pertence a família das redes neurais. Se desejável, é possível sofisticar a rede neural (e.g. utilizando mais camadas e neurônios), a fim de melhorar a previsão de perfis térmicos mais complexos. Já o AdaptableHolt não possui a mesma flexibilidade, a não ser que um outro método baseado em média exponencial seja escolhido.

Levando em conta os aspectos quantitativos e qualitativos, recomenda-se o uso do RollingPerceptorn para a tarefa de previsão de superaquecimentos.

# Capítulo 8

## Conclusão

Esta dissertação apresentou um sistema para o monitoramento térmico de *datacenters*. O projeto apresentado foi motivado pelos principais problemas relacionados à temperatura em CPDs: eficiência energética e falhas decorrentes de superaquecimento.

Apresentou-se uma rede de sensores sem-fio para o monitoramento do ambiente do *datacenter*. Tal rede é totalmente autônoma e agnóstica em relação à aplicação que roda no CPD. Sendo assim, pode-se utilizá-la em diversos cenários.

Utilizou-se um método denominado ACP para a redução dimensional da série temporal multivariada de temperatura. Através de uma técnica apresentada na literatura, o SPIRIT [40], é possível estimar os componentes principais em tempo real e viabilizar as aplicações de detecção de anomalias e previsão térmica utilizadas no trabalho.

O sistema de detecção de anomalias apresentado é robusto já que identifica quebras na correlação entre as séries temporais. Esse mecanismo foi validado através de experimentos que visavam simular ineficiências térmicas e ilhas de calor. Em ambos os casos o sistema foi capaz de detectar mudanças no perfil térmico e apontar os *motes* que geraram o aparecimento de uma nova variável oculta associada à cada anomalia.

A previsão térmica apresentada no trabalho utilizou três diferentes tipos de técnica: baseadas em suavização exponencial, autorregressão e perceptrons. Através dos experimentos realizados, comprovou-se que o previsor denominado RollingPerceptron foi capaz de apresentar a melhor relação entre falsos-positivos e tempo de antecedência (9 min).

## 8.1 Trabalhos futuros

O trabalho apresentado aqui foi guiado por um CPD pequeno, com apenas 2 bastidores e um sistema de refrigeração que não é baseado em corredores quentes e frios. Além disso, a rede de sensores utilizada nos experimentos possui apenas algumas dezenas de *motes*. Sendo assim, deseja-se realizar um experimento de maior escala com um *datacenter* maior, realizando ajustes na proposta (e.g. calibração de parâmetros) se necessário. Além disso, deseja-se obter *feedback* de operadores de *datacenter* sobre o sistema proposto a fim de avaliar melhorias que podem ser realizadas no projeto e mensurar o impacto do sistema no fluxo de trabalho desses profissionais.

A rede de monitoramento pode conter gargalos energéticos, pois o protocolo CTP não procura balancear a árvore de coleta. Uma extensão que pode ser feita é utilizar a informação da carga de bateria dos *motes* para guiar um balanceamento da árvore. Através dessa modificação, espera-se ter um tempo de vida ainda maior para a rede de monitoramento (que atualmente é de mais de 40 dias).

As redes neurais utilizadas na previsão térmica são muito simples e não possuem a capacidade de modelar sistemas não-lineares. No entanto, observou-se que até mesmo um perceptron simples, como o RollingPerceptron é capaz de superar métodos de previsão utilizados em outros trabalhos como o AR(2) [25] e Holt [43]. Existe um tipo de rede neural chamado RNN – *Recurrent Neural Network*, que é própria para a modelagem de séries temporais uma vez que possui uma memória interna das observações passadas. Deseja-se averiguar que tipo de resultado esta rede pode trazer.

Os módulos de previsão e detecção de anomalias estudados aqui podem ser integrados com o sistema TMON – *Thermal Monitor*, desenvolvido de forma colaborativa em outro trabalho [5]. Trata-se de um sistema para o monitoramento constante do CPD pelos operadores. Internamente, possui um gerenciador de eventos que pode ser usado para obter medidas da rede e gerar aletas de forma assíncrona.

Uma tendência, que vem da área de computação ubíqua, é que no futuro os ambientes serão inteligentes. Através da instrumentação de sensores no ambiente e uma interface padronizada será possível coletar uma rica quantidade de informação. Nesse contexto, o processo de detecção de anomalias poderia se valer de outros *streams* de dados e oferecer uma resposta mais precisa sobre problemas no perfil térmico do ambiente.

# Referências

- [1] ALOG. Unidades ALOG. [www.alog.com.br/infraestrutura/unidades](http://www.alog.com.br/infraestrutura/unidades) (acessado em janeiro de 2014).
- [2] APPLIED MATH MODELING, INC. CoolSim. [www.coolsimsoftware.com](http://www.coolsimsoftware.com) (acessado em janeiro de 2014).
- [3] BELL, G. C. Wireless sensors improve data center energy efficiency. Relatório Técnico, US Department of Energy, setembro 2010. [www1.eere.energy.gov/femp/pdfs/wireless\\_sensor.pdf](http://www1.eere.energy.gov/femp/pdfs/wireless_sensor.pdf) (acessado em janeiro de 2014).
- [4] BROWN, R., ET ALI. Report to congress on server and data center energy efficiency: Public law 109-431. Relatório Técnico, Lawrence Berkeley National Laboratory, agosto 2007.
- [5] BRUNO, G. Sistema para monitoramento termoenergético de CPDs. Dissertação de Mestrado, Instituto de Computação, Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2013.
- [6] BRUNO, G. Z., BOTTARI, G., GUERRA, R., LEITE, J. C. B. Building a WSN infrastructure with COTS components for the thermal monitoring of datacenters. In *ACM Symposium On Applied Computing (SAC)* (Gyeongju, Coreia do Sul, março 2014), p. 1–6. (À aparecer).
- [7] CHANDOLA, V., BANERJEE, A., KUMAR, V. Anomaly detection: a survey. *ACM Computing Surveys (CSUR)* 41, 3 (2009), 1–15.
- [8] CHLIPALA, A., HUI, J., TOLLE, G. Deluge: Data dissemination for network reprogramming at scale. Relatório Técnico, University of California at Berkeley, 2003. [www.tinyos.net/tinyos-2.x/doc/html/tep105.html](http://www.tinyos.net/tinyos-2.x/doc/html/tep105.html) (Acessado em janeiro de 2014).
- [9] CHRISTENSEN, K. Green networks: Opportunities and challenges. In *IEEE Conference on Local Computer Networks* (Zurich, Suíça, outubro 2009), p. 1–13.
- [10] DATA CENTER KNOWLEDGE. Microsoft’s Chiller-less Data Center, setembro 2009. [www.datacenterknowledge.com/archives/2009/09/24/microsofts-chiller-less-data-center](http://www.datacenterknowledge.com/archives/2009/09/24/microsofts-chiller-less-data-center) (acessado em janeiro de 2014).
- [11] FAN, X., WEBER, W.-D., BARROSO, L. A. Power provisioning for a warehouse-sized computer. In *International Symposium on Computer Architecture (ISCA)* (San Diego, California, EUA, junho 2007), p. 13–23.

- [12] FONSECA, R., GNAWALI, O., JAMIESON, K., KIM, S., LEVIS, P., WOO, A. TEP 123: collection tree protocol. Relatório Técnico, TinyOS Core Working Group, 2006. [tinynos.cvs.sourceforge.net/viewvc/tinynos/tinynos-2.x/doc/html/tep123.html](http://tinynos.cvs.sourceforge.net/viewvc/tinynos/tinynos-2.x/doc/html/tep123.html) (acessado em janeiro de 2014).
- [13] FORBES. Amazon.com Goes Down, Loses \$66,240 Per Minute, outubro 2013. [www.forbes.com/sites/kellyclay/2013/08/19/amazon-com-goes-down-loses-66240-per-minute](http://www.forbes.com/sites/kellyclay/2013/08/19/amazon-com-goes-down-loses-66240-per-minute) (acessado em janeiro de 2014).
- [14] FREITAS, L., MURTA, L., TOLEDO, R., MUNIS, T., BRAGANHOLO, V. InformeIC: Informativo do Instituto de Computação da UFF, outubro 2010. [www.ic.uff.br/index.php/pt/instituto/informe-ic](http://www.ic.uff.br/index.php/pt/instituto/informe-ic) (acessado em janeiro de 2014).
- [15] GOOGLE. Google Glass. [www.google.com.br/glass/start](http://www.google.com.br/glass/start) (acessado em janeiro de 2014).
- [16] HEATH, T., CENTENO, A. P., GEORGE, P., RAMOS, L., JALURIA, Y., BIANCHINI, R. Mercury and Freon: temperature emulation and management for server systems. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (San Jose, CA, EUA, outubro 2006), p. 106–116.
- [17] HOKE, E., SUN, J., STRUNK, J. D., GANGER, G. R., FALOUTSOS, C. InteMon: continuous mining of sensor data in large-scale self-\* infrastructures. *ACM Special Interest Group on Operating Systems (SIGOPS) 40*, 3 (julho 2006), 38–44.
- [18] INNOVATIVE-RESEARCH. TileFlow: Overview. [inres.com/products/tileflow/overview.html](http://inres.com/products/tileflow/overview.html) (acessado em janeiro de 2014).
- [19] KEOGH, E. J., LIN, J., FU, A. W.-C. Hot SAX: Efficiently finding the most unusual time series subsequence. In *IEEE international conference on Data Mining (ICDM)* (Houston, Texas, EUA, novembro 2005), p. 226–233.
- [20] KNUTH, D. E. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, Segunda Edição*. Addison-Wesley, 1981.
- [21] KOOMEY, J. Growth in data center electricity use 2005 to 2010. Relatório Técnico, Analytics Press, agosto 2011.
- [22] KUMAR, N., LOLLA, N., KEOGH, E., LONARDI, S., RATANAMAHATANA, C. A. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM International Conference on Data Mining (SDM)* (Newport Beach, CA, EUA, abril 2005).
- [23] LEVIS, P., LEE, N., WELSH, M., CULLER, D. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *ACM International Conference on Embedded Networked Sensor Systems (SenSys)* (Los Angeles, California, EUA, outubro 2003), p. 126–137.
- [24] LEVIS, P., MADDEN, S., POLASTRE, J., SZEWCZYK, R., WHITEHOUSE, K., WOO, A., GAY, D., HILL, J., WELSH, M., BREWER, E., CULLER, D. TinyOS: An operating system for sensor networks. In *IEEE International Conference on Mobile Data Management (MDM)* (Nara, Japão, maio 2006), p. 63.

- [25] LI, L., LIANG, C.-J. M., LIU, J., NATH, S., TERZIS, A., FALOUTSOS, C. Thermo-Cast: A Cyber-Physical Forecasting Model for Data Centers. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (San Diego, EUA, agosto 2011), p. 1370–1378.
- [26] LIANG, C.-J. M., LIU, J., LUO, L., TERZIS, A., ZHAO, F. RACNet: a high-fidelity data center sensing network. In *ACM International Conference on Embedded Networked Sensor Systems (SenSys)* (Berkeley, CA, EUA, novembro 2009), p. 15–28.
- [27] LIN, J., KEOGH, E., LONARDI, S., CHIU, B. A symbolic representation of time series, with implications for streaming algorithms. In *ACM SIGMOD workshop on research issues in data mining and knowledge discovery (DMKD)* (San Diego, California, EUA, junho 2003), p. 2–11.
- [28] MAIMON, O., ROKACH, L. *Data Mining and Knowledge Discovery Handbook*. Springer, 2005.
- [29] MAKRIDAKIS, S., WHEELWRIGHT, S., HYNDMAN, R. *Forecasting: Methods and Applications*. John Wiley & Sons, 1998.
- [30] MARÓTI, M., KUSY, B., SIMON, G., LÉDECZI, Á. The flooding time synchronization protocol. In *ACM International Conference on Embedded Networked Sensor Systems (SenSys)* (Baltimore, MD, EUA, novembro 2004), p. 39–49.
- [31] MARSHALL, L., BEMIS, P. Using CFD for data center design and analysis. Relatório Técnico, Applied Math Inc., janeiro 2011. [www.coolsimsoftware.com/Portals/0/PDF/WP107\\_CFD\\_for\\_Data\\_Centers.pdf](http://www.coolsimsoftware.com/Portals/0/PDF/WP107_CFD_for_Data_Centers.pdf) (acessado em janeiro de 2014).
- [32] MEMSIC. Memsic leader in MEMS Sensor Technology. [www.memsic.com](http://www.memsic.com) (acessado em janeiro de 2014).
- [33] MENTOR-GRAPHSICS. FloVENT Built Environment HVAC CFD Software. [www.mentor.com/products/mechanical/products/flovent](http://www.mentor.com/products/mechanical/products/flovent) (acessado em janeiro de 2014).
- [34] MOORE, J., CHASE, J., FARKAS, K., RANGANATHAN, P. A sense of place: Toward a location-aware information plane for data centers. Relatório Técnico, Hewlett Packard (HP), fevereiro 2004. [www.hp1.hp.com/techreports/2004/HPL-2004-27.html](http://www.hp1.hp.com/techreports/2004/HPL-2004-27.html) (acessado em janeiro de 2014).
- [35] MOORE, J. D., CHASE, J. S., RANGANATHAN, P. Weatherman: Automated, online and predictive thermal mapping and management for data centers. In *IEEE International Conference on Autonomic Computing* (Dublin, Irlanda, junho 2006), p. 155–164.
- [36] MOORE, J. D., CHASE, J. S., RANGANATHAN, P., SHARMA, R. K. Making scheduling “Cool”: Temperature-aware workload placement in data centers. In *USENIX Annual Technical Conference, General Track* (Anaheim, CA, EUA, abril 2005), p. 61–75.
- [37] MOSS, D., HUI, J., KLUES, K. TEP 105: Low power Listening. Relatório Técnico, TinyOS Core Working Group, 2007. [www.tinyos.net/tinyos-2.x/doc/html/tep105.html](http://www.tinyos.net/tinyos-2.x/doc/html/tep105.html) (acessado em janeiro de 2014).

- [38] NAVEH, A., ROTEM, E., MENDELSON, A., GOCHMAN, S., CHABUKSWAR, R., KRISHNAN, K., KUMAR, A. Power and Thermal Management in the Intel Core Duo Processor. *Intel Technology Journal* 10, 2 (maio 2006), 1–15.
- [39] NEWVILLE, M. *Least-Squares Minimization with Constraints for Python*, junho 2013. [cars9.uchicago.edu/software/python/lmfit/lmfit.pdf](http://cars9.uchicago.edu/software/python/lmfit/lmfit.pdf) (acessado em janeiro de 2014).
- [40] PAPADIMITRIOU, S., SUN, J., FALOUTSOS, C. Streaming pattern discovery in multiple time-series. In *ACM International Conference on Very Large Databases (VLDB)* (Trondheim, Noruega, agosto 2005), p. 697–708.
- [41] PEBBLE. Pebble Smartwatch. [getpebble.com](http://getpebble.com) (acessado em janeiro de 2014).
- [42] PETRUCCI, V., CARRERA, E. V., LOQUES, O., LEITE, J. C. B., MOSSÉ, D. Optimized Management of Power and Performance for Virtualized Heterogeneous Server Clusters. In *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)* (Newport Beach, CA, EUA, maio 2011).
- [43] SANTANA, C., LEITE, J. C. B., MOSSÉ, D. Load forecasting applied to soft real-time web clusters. In *ACM Symposium on Applied Computing* (Sierre, Suíça, março 2010), p. 346–350.
- [44] SHARMA, R. K., BASH, C., PATEL, C. D., FRIEDRICH, R. J., CHASE, J. S. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing* 9, 1 (2005), 42–49.
- [45] SHLENS, J. A tutorial on principal component analysis, 2005. [www.brainmapping.org/NITP/PNA/Readings/pca.pdf](http://www.brainmapping.org/NITP/PNA/Readings/pca.pdf) (acessado em janeiro de 2014).
- [46] TEIXEIRA, P. H., CLEMENTE, R., KAISER, R., VIEIRA JR, D. HOLMES: An event-driven solution to monitor data centers through continuous queries and machine learning. In *ACM International Conference on Distributed Event-Based Systems (DEBS)* (Cambridge, Reino Unido, julho 2010), p. 216–221.
- [47] WEI, L., KUMAR, N., LOLLA, V. N., KEOGH, E. J., LONARDI, S., RATANAMAHAATANA, C. A. Assumption-free anomaly detection in time series. In *International Conference on Scientific and Statistical Database Management (SSDBM)* (Santa Barbara, CA, EUA, junho 2005), p. 237–242.
- [48] YEGNANARAYANA, B. *Artificial Neural Networks*. Prentice-Hall, New Delhi, India, 2004.
- [49] ZIKANOV, O. *Essential Computational Fluid Dynamics*. John Wiley & Sons, 2010.
- [50] ZOUMBOULAKIS, M., ROUSSOS, G. Escalation: Complex event detection in wireless sensor networks. In *European Conference on Smart Sensing & Context (EuroSSC)* (Kendal, Inglaterra, outubro 2007), p. 270–285.