DETECÇÃO E LOCALIZAÇÃO DE DEGRADAÇÃO EM REDES DE ACESSO RESIDENCIAIS: APRENDIZADO A PARTIR DE RÓTULOS RUIDOSOS E DADOS REAIS NÃO ROTULADOS

Gustavo Henrique Alves dos Santos

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Rio de Janeiro
Setembro de 2026

DETECÇÃO E LOCALIZAÇÃO DE DEGRADAÇÃO EM REDES DE ACESSO RESIDENCIAIS: APRENDIZADO A PARTIR DE RÓTULOS RUIDOSOS E DADOS REAIS NÃO ROTULADOS

Gustavo Henrique Alves dos Santos

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Aprovada por: Prof. José Ferreira de Rezende
Prof. Ana Paula Couto da Silva
Prof. Antonio Jorge Gomes Abelém
Prof. Donald Fred Towsley

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2026

Santos, Gustavo Henrique Alves dos

Detecção e Localização de Degradação em Redes de Acesso Residenciais: Aprendizado a partir de Rótulos Ruidosos e Dados Reais Não Rotulados/Gustavo Henrique Alves dos Santos. – Rio de Janeiro: UFRJ/COPPE, 2026.

XII, 90 p.: il.; 29, 7cm.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2026.

Referências Bibliográficas: p. 80 – 87.

1. Network Degradation Detection. 2. Machine Learning. 3. Noisy Labels. I. de Souza e Silva, Edmundo Albuquerque *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Para Matheus e Andressa.

Agradecimentos

Gostaria de agradecer à banca pelos comentários que ajudaram a melhorar este trabalho. Gostaria também de agradecer aos meus orientadores, Edmundo e Rosa, por tantos aprendizados e por terem contribuído de maneira considerável para o meu desenvolvimento profissional durante todo o longo período em que estive no LAND.

Também gostaria de agradecer a todos os meus amigos do laboratório LAND por conta das muitas discussões e contribuições que ajudaram a construir este trabalho.

Gostaria de agradecer à minha mãe por sempre ter me incentivado aos estudos e ao Máiquel por ser um grande exemplo neste e em muitos outros sentidos. Gostaria de agradecer aos meus irmãos Cássio, Marco e Pedro por todo o companheirismo durante este e todos os outros períodos importantes da minha vida, assim como aos meus sobrinhos Ana Luísa e José Antônio, por serem fontes de inspiração e constante aprendizado.

Um agradecimento especial à pessoa que foi, sem dúvida, a mais importante para a construção desta tese, minha esposa Andressa. Sem o seu apoio, o doutorado não teria nem sido iniciado, em um período de nossas vidas em que ela teria tido todas as razões para se opor a essa decisão. Por estar sempre ao meu lado, participando das nossas conquistas, e por me dar o maior presente que eu já recebi na vida, nosso filho Matheus, serei eternamente grato.

Por fim, dedico esta tese ao meu filho Matheus Henrique, a pessoa mais importante da minha vida. Que eu possa inspirá-lo a seguir uma vida de estudo e aprimoramento pessoal como eu tive a sorte de ter tido por conta do apoio de tantas pessoas especiais.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

DETECÇÃO E LOCALIZAÇÃO DE DEGRADAÇÃO EM REDES DE ACESSO RESIDENCIAIS: APRENDIZADO A PARTIR DE RÓTULOS RUIDOSOS E DADOS REAIS NÃO ROTULADOS

Gustavo Henrique Alves dos Santos

Setembro/2026

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Programa: Engenharia de Sistemas e Computação

Detectar e localizar falhas de redes é crucial para que os provedores de serviços de Internet (ISPs) mantenham uma boa qualidade de serviço (QoS) e de experiência (QoE) dos usuários. No entanto, detectar degradações de desempenho em redes é desafiador devido à escassez de rótulos que as identifiquem. Propomos métodos baseados em aprendizado de máquina capazes de detectar degradações em redes quando rótulos confiáveis são escassos ou inexistentes. Para o cenário em que nenhum rótulo está disponível, elaboramos duas técnicas baseadas em modelos de aprendizado de máquina não supervisionados: uma abordagem baseada em decomposição tensorial, capaz de identificar degradações usando séries temporais multivariadas e uma técnica baseada em detecção de pontos de mudança capaz de localizar regiões que experimentam degradações. O segundo cenário inclui a utilização de chamados ao *call-center* do ISP, que apresentam ruído inerente ao comportamento dos usuários. Desenvolvemos um novo método de aprendizado de máquina estatístico que correlaciona medições de QoS e rótulos de QoE com ruído para detectar e localizar degradações. A técnica proposta é capaz de ranquear as detecções, auxiliando o ISP a priorizar esforços operacionais durante o *troubleshooting*. A eficácia dos métodos desenvolvidos é demonstrada utilizando um conjunto de dados rico, com granularidade fina, coletado por milhares de roteadores residenciais durante dezenas de meses, obtido em parceria com um ISP de médio porte e uma *startup*. Nossos resultados indicam que as técnicas propostas são capazes de detectar e localizar eventos reais de rede que podem afetar a QoE dos usuários do ISP.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

DETECTING AND LOCALIZING NETWORK DEGRADATION IN RESIDENTIAL ACCESS NETWORKS: LEARNING FROM NOISY LABELS AND UNLABELED REAL-WORLD DATA

Gustavo Henrique Alves dos Santos

September/2026

Advisors: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Department: Systems Engineering and Computer Science

Detecting and localizing network faults is crucial for Internet Service Providers (ISPs) to maintain high quality of service (QoS) and quality of experience (QoE) for users. However, detecting network performance degradations is challenging because labels are scarce. We propose machine learning-based methods that can detect network degradations when reliable labels are scarce or nonexistent. When no labels are available, we devised two unsupervised machine learning techniques: a tensor decomposition-based approach that identifies degradations using multivariate time series, and a change-point detection technique that localizes regions experiencing degradations. The second scenario uses information from calls to the ISP's call center, which contain noise inherent to user behavior. We developed a statistical machine learning method that correlates QoS measurements and noisy QoE labels to detect and localize degradations. The proposed technique can rank detections, helping the ISP prioritize troubleshooting efforts. We show the effectiveness of the developed methods using a rich, fine-grained dataset collected by thousands of residential routers over many months, obtained through a partnership with a medium-sized ISP and a startup. Our results show that the proposed techniques can detect and localize real network events that may affect the ISP users' QoE.

Sumário

Lista de Figuras **x**

Lista de Tabelas **xii**

1 Introdução **1**

1.1 Trabalhos publicados 4

1.2 Organização do trabalho 4

2 Revisão Bibliográfica **5**

2.1 Medições passivas 6

2.2 Medições em datacenters 7

2.3 Tomografia de rede 9

2.4 Grafo de dependências 12

2.5 Aprendizado de máquina 14

2.6 Contribuições de nossa proposta 16

3 Background **19**

3.1 PARAFAC 19

3.2 Hidden Markov Models 22

4 Metodologia **25**

4.1 Framework 25

4.2 Construção do Dataset 28

4.2.1 Infraestrutura de Medição 28

4.2.2 Dados disponibilizados pelo ISP 29

4.3 Resumo 30

5 Métodos Não Supervisionados **32**

5.1 Método baseado em Decomposição Tensorial 32

5.1.1 Pré-processamento 33

5.1.2 Treinamento do modelo 33

5.1.3 Detecção de degradação 34

5.1.4 Correlação Espaço-Temporal 35
5.2 Método baseado em Detecção de Pontos de Mudança 35
5.2.1 Pré-processamento 36
5.2.2 Treinamento do modelo 36
5.2.3 Detecção de degradação 37
5.2.4 Correlação Espaço-Temporal 39
5.3 Resumo 40

6 Método Baseado em Rótulos com Ruído 41
6.1 Pré-processamento 42
6.2 Treinamento do modelo 43
6.2.1 Descrição do modelo 43
6.2.2 Estimativa dos parâmetros do modelo 46
6.3 Detecção de degradação 48
6.4 Correlação Espaço-Temporal 48
6.5 Ranqueamento de degradações 52
6.6 Resumo 53

7 Resultados 55
7.1 Métodos não supervisionados 55
7.1.1 Método baseado em Decomposição Tensorial 55
7.1.2 Resultados do método baseado em Detecção de Pontos de Mudança 61
7.2 Método baseado em rótulos com ruído 65
7.2.1 Modelo de detecção baseado em rótulos com ruído 66
7.2.2 Resultados da correlação espaço-temporal 68
7.2.3 Resultados do método de ranqueamento de degradações 73
7.2.4 Eventos de rede detectados 74

8 Conclusão 78

Referências Bibliográficas 80

A Resultados Auxiliares 88
A.1 Algoritmo de Expectation-Maximization 88

Lista de Figuras

2.1 Categorização dos trabalhos relacionados 6

3.1 Hidden Markov Models representadas através de um Probabilistic Graphical Model 23

4.1 Visão geral da abordagem proposta 26
4.2 Infraestrutura de medição 28
4.3 Partição da rede em regiões 30

5.1 Aplicação do método de votação majoritária. Rótulos vermelhos indicam degradação de desempenho detectada. 40

6.1 *Probabilistic Graphical Model* representando os relacionamentos entre as variáveis aleatórias propostas 44
6.2 Exemplo de correlação espaço-temporal ($\delta = 0.45$, $\eta = 0.95$). Nó C apresenta problemas e é responsável pela degradação do desempenho observada. 52

7.1 Séries temporais sumarizadas por cluster 56
7.2 Exemplo de correlação espaço-temporal (amostras faltantes são representadas utilizando quadrados vermelhos). 58
7.3 Desempenho da rede obtido a partir da clusterização dos resíduos . . 59
7.4 Eventos detectados pelo nosso método 60
7.5 Resultados da clusterização entre os dias 01/03/2020 e 31/07/2020. O primeiro dia rotulado representa o início do período de confinamento enquanto os dias seguintes representam o início de diferentes fases de flexibilização da quarentena. 61
7.6 Relação entre clusters obtidos e chamados ao call center. 62
7.7 Modelo de Markov oculto obtido após o treinamento 63
7.8 Correlação espaço-temporal aplicada nos dias 08/08/2018 (esq.) e 23/09/2018 (dir.). Nós em vermelho experimentam degradação de performance. 64

7.9 Mudança de comportamento da rede de usuário após intervenção na rede do ISP. Resultados em vermelho são associados a estados ocultos em $\mathcal{R}$ 64
7.10 Fração de amostras das séries temporais associadas a estados em $\mathcal{R}$. Estados em $\mathcal{R}$ são representados pela cor vermelha. 65
7.11 Taxa de acerto para aplicação de Viterbi *online* 66
7.12 Implementação de prova de conceito. Marcadores indicam localização de roteadores e cor do marcador indica a QoS da rede identificada pelo método 67
7.13 Modelo obtido para $Y = 0$. $P[Y = 0] = 0.8208$ e $P[\tilde{Y} = 1|Y = 0] = 0.0026$ 68
7.14 Modelo obtido para $Y = 1$. $P[Y = 1] = 0.1792$ e $P[\tilde{Y} = 1|Y = 1] = 0.0116$ 68
7.15 Comparação dos modelos de detecção utilizando votação majoritária (Intervalo de Confiança de 95%) 71
7.16 Comparação dos métodos de correlação espaço-temporal utilizando modelo baseado em rótulos com ruído (Intervalo de Confiança de 95%) 71
7.17 Comparação entre $\hat{p}_b$ e $\hat{p}_g$ no conjunto de teste (Intervalo de Confiança de 95%) 72
7.18 95% HPD da distribuição da diferença entre p_b e p_g para cada mês . . 73
7.19 Avaliação de desempenho do método de ranqueamento 74
7.20 Evolução temporal do ranqueamento durante uma semana filtrando regiões com ao menos 2 dias no top 5 75
7.21 Exemplos de degradações reais detectadas pelo método 77

Lista de Tabelas

2.1 Comparação entre os trabalhos relacionados e as técnicas propostas . 18

6.1 Notação: Índices, conjuntos, dimensões e variáveis aleatórias 46
6.2 Notação: Parâmetros do modelo. 46
6.3 Notação: Variáveis auxiliares 47
6.4 Notação: Correlação Espaço-Temporal. 51

Capítulo 1

Introdução

Detectar degradações de desempenho de rede é essencial para que provedores de serviço de internet (ISPs) sejam capazes de oferecer uma boa qualidade de serviço (QoS) e, consequentemente, uma boa qualidade de experiência (QoE) para os seus clientes. No nosso trabalho, utilizamos o termo *degradação de rede* para denotar uma deterioração persistente ou operacionalmente relevante em uma ou mais métricas de desempenho. Este é um conceito mais específico do que uma anomalia de rede, que pode se referir a comportamentos estatisticamente fora do esperado, incluindo desvios de curta duração que podem não produzir uma degradação significativa na qualidade do serviço. Muitos trabalhos da literatura lidam com o problema de detecção de anomalias [1], e manteremos essa terminologia quando adotada pelos seus autores.

A criação de métodos para detectar degradação de rede apresenta diversos desafios. Uma das principais dificuldades consiste na obtenção de rótulos identificando esses eventos. Para lidar com a falta de *labels*, diversos trabalhos recorrem à criação manual de rótulos a partir da inspeção visual de certas métricas de rede [2–4], um processo custoso e suscetível a erros. A avaliação manual depende do conhecimento específico de *experts* e é inerentemente subjetiva [1]. Além do mais, eventos de degradação não previstos podem não ser identificados a partir de métodos manuais. Outros trabalhos recorrem à utilização de dados sintéticos para avaliar modelos teóricos [5–8], utilizando cenários artificiais que podem não representar diversas características da realidade, como a correlação temporal do desempenho da rede ou a estrutura e a escala de topologias de redes reais.

A informação de chamados ao *call center* realizados por usuários de um provedor de rede (ISP) pode ser utilizada para auxiliar a detecção e localização de problemas na rede. A existência de um chamado frequentemente representa uma degradação na qualidade da experiência percebida por um usuário, uma informação valiosa para o ISP. Além disso, uma base de dados com informação de chamados pode ser obtida sem a necessidade de esforço extra para os ISPs. No entanto, a utilização

de chamados para a criação de modelos apresenta diversos desafios. Um usuário pode não criar um chamado quando o desempenho da rede é degradado por conta do tempo perdido durante o atendimento ou por não ter utilizado a rede durante uma falha. Por outro lado, um cliente do ISP pode realizar um chamado quando o problema não é relacionado à sua rede de acesso, como em casos de falha de um dispositivo residencial ou de problemas de conectividade em redes sem fio. Desta forma, a informação provida pela existência de chamados é ruidosa, o que torna a sua utilização para a criação de modelos desafiadora.

Nosso trabalho tem como objetivo criar métodos capazes de extrair padrões de dados reais com o intuito de possibilitar a detecção e localização de eventos que causam degradação de desempenho em redes. Nosso intuito é criar um método prático para detecção e localização automática de degradações de desempenho que possa auxiliar provedores residenciais a resolverem problemas de maneira proativa com menor tempo de resposta. Com esta finalidade, propomos um framework baseado em modelos de aprendizado de máquina para detecção de degradação capaz de lidar com as características observadas em medições reais, como a presença de dados faltantes e a correlação espaço-temporal existente nas amostras.

O framework proposto é empregado para construir técnicas capazes de lidar com a escassez de rótulos confiáveis. Propomos técnicas não supervisionadas para lidar com cenários em que nenhum rótulo está disponível. Além disso, consideramos o conceito de rótulos com ruído, definido na literatura de inteligência artificial [9], para propor uma nova abordagem, baseada em um modelo estatístico de detecção de degradação, capaz de utilizar a informação proveniente de chamados ao *call center* levando em consideração o ruído inerente ao comportamento dos usuários.

Um pré-requisito para alcançar os objetivos propostos é a obtenção de um conjunto de dados reais coletado por um número suficientemente grande de usuários durante uma quantidade de tempo relevante. A partir de uma parceria entre a universidade, uma *startup* iniciada na incubadora da universidade e um ISP de médio porte, fomos capazes de obter um conjunto de dados de medições de QoS rico e com granularidade fina. O conjunto de dados contém o resultado de medições de métricas como latência, perda de pacotes e tráfego com granularidade de minuto, obtido a partir de milhares de medidores espalhados pela rede do provedor, através do qual é possível extrair um conjunto de séries temporais de métricas de QoS. Também tivemos acesso à topologia da rede do provedor e a um banco de dados de chamados realizados ao *call center*, o que possibilitou a criação de métodos para a localização de degradações de desempenho que afetam a qualidade da experiência dos usuários.

A partir dos resultados do modelo de detecção de degradação e da localização dos medidores na rede do ISP, torna-se possível correlacionar os resultados de medidores próximos geograficamente com o intuito de localizar quais regiões da rede são

potencialmente afetadas por uma degradação de desempenho. Desenvolvemos neste trabalho novos métodos de correlação espacial para identificar quais equipamentos são potencialmente responsáveis por uma falha observada. Por fim, para auxiliar o ISP a priorizar esforços durante o processo de *troubleshooting*, criamos um novo método de ranqueamento que utiliza os resultados da correlação espaço-temporal para ordenar regiões potencialmente afetadas por uma degradação de desempenho de acordo com o grau de confiança na detecção estimado utilizando inferência bayesiana.

Em resumo, é essencial que ISPs realizem a detecção automática de problemas na rede com o objetivo de prover uma boa qualidade de serviço. A detecção de mudanças estatísticas em métricas de QoS extraídas a partir de medições ativas pode permitir a detecção automática da degradação de desempenho da rede e a localização da região afetada. No entanto, a dificuldade de obtenção de um conjunto de dados detalhado, a ausência de rótulos confiáveis identificando degradação de desempenho e o fato de as séries temporais de medições serem não estacionárias são grandes desafios enfrentados pelos métodos existentes. Nosso trabalho tem como objetivo resolver o problema com modelos adequados capazes de extrair padrões automaticamente a partir de conjuntos de dados reais. Nossas principais contribuições incluem:

- Criação de modelos de detecção de degradação de desempenho a partir de séries temporais de medições de QoS. Os métodos propostos levam em consideração a correlação temporal entre as amostras, algo menos explorado na literatura (como detalhamos no Capítulo 2). Para o cenário em que *labels* identificando degradações não estão disponíveis, consideramos modelos não supervisionados. Por outro lado, para o cenário em que temos acesso a rótulos não confiáveis, desenvolvemos um novo modelo estatístico capaz de lidar com rótulos com ruído.

- Elaboração de métodos para localização de regiões da rede que experimentam degradação de desempenho a partir da correlação espacial dos resultados do modelo de detecção.

- Criação de um novo método para identificar degradações que devem ser prioritariamente analisadas com o objetivo de auxiliar o ISP durante o processo de *troubleshooting*.

- Aplicação dos métodos propostos a um conjunto de dados real de granularidade fina coletado em milhares de residências por diversos meses para mostrar a efetividade do método. Mostramos a viabilidade da nossa abordagem a partir

da implementação de uma prova de conceito da técnica para detecção e localização de degradações de maneira *online* em um ambiente real com milhares de medidores.

1.1 Trabalhos publicados

- O novo método não supervisionado de detecção de degradação baseado em decomposição tensorial (Seção 5.1) foi publicado em [10–12]
- O novo método não supervisionado de detecção e localização de degradação baseado na detecção de pontos de mudança (Seção 5.2) foi publicado em [13]
- O novo método para detecção e localização de degradação de desempenho baseado em métricas de QoS e rótulos de QoE com ruído (Capítulo 6) foi publicado em [14]

1.2 Organização do trabalho

O trabalho é organizado como se segue. O Capítulo 2 analisa os trabalhos relacionados e mostra como a nossa proposta se diferencia da literatura. O Capítulo 3 apresenta os conceitos básicos utilizados pelas técnicas consideradas neste trabalho. O Capítulo 4 descreve o framework proposto para detecção e localização de degradação de desempenho e o conjunto de dados considerado. O Capítulo 5 descreve os métodos não supervisionados, enquanto o Capítulo 6 detalha a técnica baseada em rótulos de QoE com ruído. O Capítulo 7 apresenta os resultados obtidos. Por fim, o Capítulo 8 apresenta a conclusão e descreve trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo, realizamos uma análise dos trabalhos da literatura relacionados à detecção e localização de eventos de rede. Discutimos as limitações das abordagens anteriores e destacamos as principais contribuições do nosso trabalho em relação à literatura.

Diversos trabalhos relacionados formulam o problema como detecção de anomalias em redes [15–18]. Ao descrever esses trabalhos, mantemos a terminologia adotada por seus autores. Para os métodos propostos nesta tese, utilizamos o termo *detecção de degradação de desempenho de rede*, conforme definido no Capítulo 1.

A Figura 2.1 apresenta a categorização proposta para trabalhos relacionados à detecção de falhas em redes. Nosso trabalho é baseado nos resultados de medições ativas de rede, o que é complementar a trabalhos baseados nos resultados de medições passivas, como discutimos na Seção 2.1. O foco da nossa abordagem é detectar e localizar falhas em redes de provedores de serviço residenciais, diferentemente de trabalhos da literatura que se concentram em redes de datacenters. Na Seção 2.2, descrevemos como as suposições adotadas por métodos projetados para datacenters não se aplicam ao cenário residencial.

As técnicas propostas neste trabalho são baseadas em Aprendizado de Máquina. Descrevemos na Seção 2.3 as vantagens da nossa abordagem em relação a trabalhos anteriores baseados em tomografia de redes. Apresentamos como a nossa proposta se compara com abordagens baseadas em grafos de dependências na Seção 2.4. Analisamos outros trabalhos que também adotam técnicas baseadas em Aprendizado de Máquina, detalhando as contribuições da nossa proposta em relação à literatura na Seção 2.5. Por fim, resumimos na Seção 2.6 as principais contribuições do trabalho proposto.

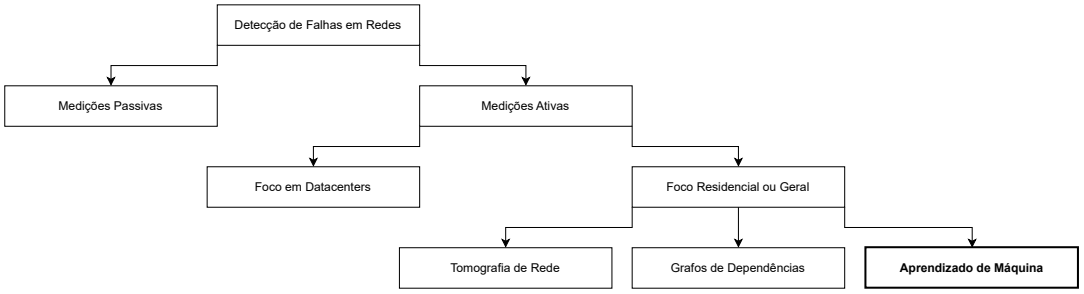


Figura 2.1: Categorização dos trabalhos relacionados

2.1 Medições passivas

Muitos trabalhos da literatura realizam a detecção de anomalias a partir de métricas relacionadas ao volume de tráfego, como número de bytes/pacotes transmitidos entre pares Origem-Destino [2–4, 8, 15, 19–21]. Dentre estes trabalhos, diversas técnicas são baseadas em decomposição matricial [2, 15] ou decomposição tensorial [8, 19–21] aplicadas a medições de volume de tráfego. Consideramos que estas técnicas são complementares à proposta deste trabalho: medições passivas de tráfego podem ser obtidas com menor *overhead*, mas medições ativas são mais eficientes para detectar certos tipos de falhas que não são detectadas de maneira passiva, como falhas silenciosas ou *gray failures* [22].

Outros trabalhos utilizam o resultado de medições passivas de latência ou perda de um serviço de internet para realizar a detecção e localização de falhas na rede. O trabalho de PADMANABHAN *et al.* [23] utiliza o resultado de medições passivas de perda, extraídas a partir do tráfego entre clientes e um servidor web, e a topologia da rede como entradas para um método de tomografia binária que identifica links com taxa de perda acima de um determinado limiar. Desvantagens deste método incluem o alto custo computacional e a esparsidade dos resultados de medições passivas.

Os autores de [24] propõem o framework CEM (Crowdsourcing Event Monitoring) para detectar e localizar falhas a partir de medições passivas de uma aplicação de BitTorrent em dispositivos finais. Nesta abordagem, os resultados de detecção local de cada usuário são correlacionados e um método probabilístico diferencia problemas individuais de falhas na rede afetando múltiplos usuários. A utilização de medições provenientes de dispositivos finais pode prejudicar o desempenho do método proposto, uma vez que diversos fatores de confusão, como carga no dispositivo e qualidade da rede de acesso sem fio, podem alterar o resultado de medições [25].

O método Argus é proposto por YAN *et al.* [16] com o objetivo de detectar falhas a partir de medições passivas em serviços gerenciados por um ISP. Com este objetivo, o método agrega espacialmente resultados de medição e extrai uma série temporal sumarizada em cada região considerada. Em seguida, o algoritmo de Holt-

Winters [26] é empregado para detectar anomalias a partir do residual do modelo. A sumarização dos resultados de medições pode suavizar comportamentos anômalos, prejudicando a sua detecção. Além disso, o método proposto possui um número considerável de hiperparâmetros de difícil ajuste.

A efetividade de métodos baseados em monitoração passiva é limitada pela esparsidade das medições [16], uma vez que medições passivas de tráfego dependem do padrão de utilização de usuários e da popularidade da aplicação. Além disso, medições passivas realizadas em dispositivos finais precisam lidar com diferentes fatores de confusão, como qualidade da rede sem fio e carga no dispositivo. Consideramos no nosso trabalho medições ativas coletadas com granularidade fina de maneira contínua em roteadores residenciais, a partir das quais obtemos um conjunto de dados robusto em que controlamos fatores de confusão associados ao tráfego residencial e às redes sem fio.

2.2 Medições em datacenters

Diversas abordagens para a detecção e localização de falhas em datacenters podem ser encontradas na literatura recente. Estes métodos levam em consideração características específicas de redes de datacenters para prover métodos capazes de detectar problemas de maneira automática.

O trabalho de HERODOTOU *et al.* [22] propõe um método para detecção e localização de falhas a partir da combinação de medições ativas e passivas. O algoritmo proposto leva em consideração as múltiplas rotas existentes entre servidores de um datacenter através de um modelo probabilístico, formulando a localização de falha de componentes da rede como um problema de otimização baseado na suposição de que no máximo um componente apresenta falha em cada caminho de medição. De maneira similar, o framework Themis [27], proposto para detectar e localizar falhas de rede a partir da combinação de medições ativas e passivas, também é focado no cenário em que um único link apresenta falha. A suposição de falha única em cada caminho de medição pode não ser realista em datacenters de larga escala e pode causar um grande número de falsos positivos e falsos negativos [28].

O sistema de monitoramento Pingmesh é proposto por GUO *et al.* [29] com o objetivo de analisar a latência observada por aplicações e serviços hospedados em um datacenter. O método realiza medições contínuas a partir de todos os servidores de um mesmo datacenter e entre uma fração de pares de servidores de diferentes datacenters, baseando-se na suposição de que a rede de um datacenter possui alta capacidade para lidar com o *overhead* associado. Falhas são detectadas a partir de *thresholds* aplicados aos resultados de medições. O trabalho não propõe um

algoritmo para localização de falhas e pode gerar um grande *overhead* quando a suposição de que a rede possui alta capacidade é violada.

O trabalho de PENG *et al.* [30] propõe o método deTector para realizar a detecção e localização de falhas em redes de datacenters. O método propõe um algoritmo guloso para a escolha dos caminhos de medição com o objetivo de diminuir o *overhead* do monitoramento. Para lidar com o alto custo computacional do algoritmo, os autores se baseiam na suposição de que a topologia de datacenters apresenta simetria. A detecção de uma falha é realizada a partir de *thresholds* aplicados aos resultados de medições, enquanto a localização dos componentes responsáveis pela falha utiliza uma abordagem gulosa similar à adotada pelo algoritmo Tomo [31]. O método é avaliado considerando apenas simulações e falhas emuladas, e a ineficiência computacional do algoritmo pode dificultar a aplicação do método em cenários reais.

Os autores de [32] propõem o sistema Simon para inferir variáveis que identificam o estado da rede de um datacenter, como o tamanho das filas de pacotes e a utilização de cada link. O método assume que os *hosts* responsáveis por medições estão sincronizados e apresentam recursos computacionais suficientes para realizar o monitoramento dos timestamps de envio e recepção de cada pacote em tempo real. O sistema Zoom2Net [33] é baseado na aplicação de um modelo transformer para inferir o comportamento de uma série temporal com granularidade fina a partir de medições realizadas com granularidade mais grossa com o objetivo de diminuir o custo associado à coleta de dados. No entanto, a técnica necessita de um dataset de granularidade fina coletado de maneira mais custosa. O framework μMon [34] também propõe uma abordagem baseada na captura de pacotes para detectar eventos de congestionamento com granularidade de microssegundos. Métodos baseados em monitoramento de pacotes podem ser computacionalmente inviáveis para medições em redes residenciais. Em especial, nossa infraestrutura de medição é baseada em roteadores residenciais de baixo custo e baixo poder computacional.

O trabalho de TAN *et al.* [28] propõe o método NetBouncer, um sistema de localização de falhas em redes de datacenters a partir de medições ativas. O método se baseia na suposição de que a rede é saudável para realizar a identificação do estado de cada link da rede através de um sistema de equações relacionando o resultado das medições fim a fim com a taxa de perda de cada link. Caso um dispositivo não faça parte do caminho de nenhuma medição com bom desempenho (isto é, sem perda), o dispositivo é considerado falho. Considera-se que um link apresenta falha quando a perda estimada apresenta um valor maior do que um *threshold*. A abordagem proposta não considera a correlação temporal entre amostras e possui um método de medição com alto *overhead*.

Métodos considerados nesta seção são criados sob medida para a detecção e/ou localização de falhas em redes de datacenters. Consequentemente, diversas supo-

sições sobre o comportamento e/ou estrutura da rede são levadas em consideração durante a construção destes métodos, incluindo a existência de caminhos redundantes entre pares de *hosts*, a alta capacidade de transmissão da rede, a simetria da topologia e a suposição de que a rede é saudável e apresenta poucas falhas. Muitas destas suposições não se traduzem para o contexto de redes de acesso residencial, o que inviabiliza a utilização destes métodos no cenário considerado pelo nosso trabalho.

2.3 Tomografia de rede

Muitos trabalhos consideram o problema de tomografia de rede, cujo objetivo é inferir o desempenho de componentes presentes no interior da rede a partir de medições fim a fim [5]. Abordagens iniciais utilizam medições multicast para inferir o comportamento de métricas de rede em cada link levando em consideração a correlação a nível de pacote. CÁCERES *et al.* [35] desenvolve estimadores de máxima verossimilhança para a taxa de perda a partir dos resultados de medições multicast. PRESTI *et al.* [36] estende o trabalho para o cálculo do atraso variável de pacotes, isto é, o atraso obtido excluindo o tempo de propagação. Ambos os trabalhos se baseiam no pressuposto de que pacotes apresentam correlação perfeita por conta da utilização de medições multicast. Além disso, os trabalhos são avaliados utilizando simulações em cenários artificiais que não representam ambientes reais.

Uma extensão da tomografia de rede para medições unicast é proposta em DUFFIELD *et al.* [6]. Os autores argumentam que pacotes multicast podem observar condições de rede distintas das observadas por pacotes unicast e propõem a utilização de um trem de pacotes unicast back-to-back com o objetivo de se aproximar da correlação perfeita observada em medições multicast. Limitações do trabalho incluem a utilização de simulações baseadas em cenários artificiais para a avaliação do método e a ausência da correlação temporal entre amostras na modelagem.

O trabalho de GHITA *et al.* [7] propõe um método baseado em sistemas de equações para inferir a taxa de perda de pacotes em cada link a partir de medições fim a fim. Os autores aplicam o método de eliminação Gauss-Jordan para encontrar um sistema de equações linearmente independentes que pode ser resolvido de maneira única. Assume-se que os links que não fazem parte das equações linearmente independentes possuem perda zero, o que possibilita uma estimativa para a taxa de perda de toda a rede. O desempenho do algoritmo pode ser degradado quando esta suposição é violada e o método não considera a correlação temporal entre amostras observada em medições reais.

Estes trabalhos não consideram as possíveis variações temporais das distribuições

de métricas de rede. Nossos dados mostram que mudanças na distribuição das métricas podem ocorrer dentro de um mesmo dia, o que motiva a utilização de um método capaz de detectar pontos de mudança estatística (Seção 5.2). A avaliação dos métodos também é simplificada, utilizando simulações com taxas de perda fixas, muitas vezes considerando topologias artificialmente geradas. Nossa abordagem baseada em aprendizado de máquina é capaz de aprender a distribuição das métricas de rede utilizando dados reais e levando em consideração a possibilidade de variações temporais.

Tomografia binária de rede

O trabalho de DUFFIELD [5] propõe o conceito de tomografia binária de rede, uma versão modificada do problema de tomografia de rede original. Nesta técnica associa-se a cada link um estado ("bom" ou "ruim") de acordo com o valor de alguma estatística extraída a partir de uma métrica de rede (exemplos incluem taxa de perda e latência média). Um link é considerado "bom" quando esta estatística se encontra abaixo de um determinado limiar escolhido pelo operador, sendo considerado "ruim" caso contrário. O objetivo das técnicas de tomografia binária de rede é identificar o estado de cada link da rede a partir dos resultados de medições fim a fim, identificando quais componentes da rede apresentam degradação de desempenho.

Os autores de [23] propõem métodos de otimização e estatística para identificar links com degradação de desempenho a partir de medições passivas. Um link é considerado "ruim" quando a taxa de perda é maior do que um limiar escolhido pelo operador. A escolha do limiar para definir o que representa um link "ruim" é empírica e pode não representar de maneira adequada o impacto na experiência do usuário.

Em [5], um método simples para a detecção de links "ruins" denominado *Smallest Consistent Failure Set* (SCFS) é proposto. O algoritmo atribui a degradação de desempenho observada em medições fim a fim ao menor conjunto possível de links "ruins" necessários para explicar todas as degradações observadas, baseando-se na suposição de que falhas são raras e ocorrem de maneira independente entre links. A violação destas suposições degrada os resultados obtidos pelo algoritmo. Além disso, a avaliação do método considera apenas simulações baseadas em cenários artificiais e não leva em conta a correlação temporal das amostras.

O trabalho de DHAMDHERE *et al.* [31] propõe a técnica NetDiagnoser, uma extensão do trabalho de DUFFIELD [5] para lidar com o cenário multi-AS (Autonomous System), com o objetivo de identificar a localização de falhas persistentes que causam problemas de conectividade na rede. O método propõe uma abordagem gulosa para identificar os links responsáveis por uma degradação, em que a cada

passo seleciona-se o conjunto de links que justifica a maior quantidade de falhas ainda não explicadas. O método proposto se limita a detectar falhas persistentes e não é capaz de lidar com falhas transientes, como congestionamentos na rede.

Os autores de [37] propõem um método para a solução do problema de tomografia binária a partir da utilização de álgebra booleana. O método estima a probabilidade *a priori* para o estado de cada link e combina esta probabilidade com o resultado de medições para identificar links com degradação de desempenho. A identificação é feita utilizando um algoritmo heurístico baseado em um estimador MAP (*Maximum a Posteriori*). O trabalho não considera a variação temporal das métricas de rede observada em redes reais, assumindo que as probabilidades *a priori* são mantidas durante o período de medição.

Técnicas para melhorar o desempenho de algoritmos de tomografia binária são propostas por CUNHA *et al.* [38]. Os autores consideram o problema de detecção de *black holes*, definidos como falhas persistentes que causam problemas de conectividade e que não podem ser solucionadas de maneira automática pela rede. O trabalho propõe métodos de confirmação de falha e de agregação de resultados de medições para melhorar a qualidade da localização através de tomografia binária. O trabalho proposto lida apenas com falhas persistentes, não sendo apropriado para a detecção de falhas transientes.

Os autores de [17] propõem um método de detecção e localização de links com falhas persistentes que tem como objetivo permitir um *tradeoff* entre o *overhead* gerado por medições e a frequência com a qual cada link é monitorado. O algoritmo assume que falhas são raras e que apenas um único link apresenta falha em um determinado instante de tempo. Baseada nesta suposição, a técnica identifica de maneira empírica qual o link responsável por uma degradação de desempenho a partir do envio de *probes* extras para isolar o link anômalo. O método pode apresentar desempenho ruim quando a suposição de anomalia única e persistente é violada e a correlação temporal entre amostras não é considerada.

O trabalho de GHITA *et al.* [39] mostra que técnicas de tomografia binária podem apresentar desempenho inferior quando condições que podem ser observadas em dados reais, como dependência entre o estado dos links, probabilidade de falha dos links não homogênea e variação dos estados da rede no tempo contradizem suposições utilizadas pelos métodos de detecção e localização de falhas. Os autores propõem um método que possui como objetivo resolver um problema mais simples, isto é, estimar a frequência de congestionamento dos links, sem identificar quais links apresentam falha em cada instante de tempo. Os autores argumentam que esta versão simplificada do problema pode ser resolvida de maneira eficiente com a utilização de suposições mais fracas. Consequentemente, o método identifica apenas quais links apresentam congestionamento com maior frequência e não considera de

maneira explícita a correlação temporal entre amostras.

O método Hermes é proposto em [40] com o objetivo de detectar e localizar eventos de rede a partir dos resultados de testes de velocidade do M-Lab em conjunto com medições de traceroute bidirecionais, utilizadas para inferir a topologia da rede. A técnica detecta eventos utilizando testes estatísticos aplicados à latência e à vazão medidas em um determinado dia, em comparação com um baseline extraído a partir dos resultados de 7 dias anteriores. A localização dos eventos é realizada através da aplicação de duas técnicas complementares: uma técnica denominada *Temporal Tomography*, apropriada para localizar mudanças de rota, e uma técnica denominada *Correlation Tomography*, que tem como foco localizar congestionamentos. O método possui como principais limitações o ruído associado às medições realizadas sob demanda por usuários, que geram uma amostra com resultados potencialmente impactados por fatores de confusão, como redes sem fio e tráfego concorrente, e cuja cobertura da rede depende de padrões de uso, além do ruído associado à medição da topologia via traceroute. O método também não utiliza a perda de pacotes como um sinal primário de detecção por conta dos fatores de confusão associados às medições realizadas em dispositivos finais. Por fim, a técnica proposta não modela explicitamente a correlação temporal existente entre as métricas de rede.

As abordagens descritas nesta seção compartilham algumas limitações. As técnicas consideradas utilizam um limiar empírico para definir o estado de um componente de rede ("bom" ou "ruim") ou resolvem uma versão simplificada do problema, em que o único objetivo é identificar problemas de conectividade. Os métodos não consideram a variação temporal dos estados de cada componente da rede observada em dados reais. Os trabalhos muitas vezes consideram para avaliação a simulação e emulação de falhas, além de experimentos simplificados sem *ground truth*. A ausência de datasets reais com granularidade fina impossibilita a estimativa de modelos capazes de levar em consideração a correlação temporal entre amostras. Por fim, muitos trabalhos possuem detecção limitada a certas condições, como falha única e/ou persistente.

2.4 Grafo de dependências

Outra abordagem utilizada para localização de falhas em componentes de rede é baseada no conceito de grafo de dependências. Este grafo tem como objetivo representar as dependências existentes entre os diferentes componentes da rede de forma a possibilitar a associação de um sintoma de falha observado a possíveis causas latentes. Como exemplo, é possível utilizar grafos de dependências para associar uma falha em um link IP detectada por SNMP (sintoma) com a ruptura de uma fibra

óptica (causa).

O método Shrink [41] realiza a localização da causa de uma falha em um link IP detectada a partir de mensagens SNMP. Para realizar a localização, o algoritmo utiliza como entrada um grafo bipartido de dependências, associando links IP lógicos a componentes físicos. Uma rede bayesiana, criada a partir do grafo de dependências, é utilizada para identificar o conjunto de falhas em componentes físicos que possui maior probabilidade de explicar falhas observadas em links IP. Para permitir uma detecção eficiente, os autores assumem que no máximo κ falhas ocorrem de maneira simultânea. A montagem do grafo de dependências depende de uma base de dados relacionando componentes físicos e links IP lógicos que pode ser de difícil obtenção e manutenção.

O trabalho de BAHL *et al.* [42] propõe a técnica Sherlock para identificar equipamentos ou serviços responsáveis por degradação de desempenho. O método infere um grafo de dependências multinível a partir de medições passivas e da topologia da rede estimada utilizando traceroute. O algoritmo de detecção de causa raiz proposto utiliza o grafo de dependências para avaliar os possíveis estados associados a cada nó do grafo e retorna as causas mais prováveis para uma falha, baseando-se na suposição de que no máximo k falhas existem de maneira simultânea. A montagem do grafo de dependências depende dos padrões de utilização dos usuários da aplicação e o algoritmo possui uma grande quantidade de parâmetros cuja definição não é trivial.

Os autores de [43] propõem a utilização de modelagem de risco para a localização de falhas detectadas através de medições ativas ou passivas. O método utiliza para localização um grafo de dependências bipartido que modela a dependência entre sintomas observados e suas possíveis causas, denominadas riscos compartilhados. A partir do grafo de dependências e dos resultados de medições, os autores propõem uma heurística gulosa para localização que identifica o menor conjunto de componentes com falhas capaz de explicar as degradações de desempenho observadas. A criação dos grafos de dependência é complexa, uma vez que os grafos são dinâmicos e podem representar um grande número de componentes. Além disso, o algoritmo possui diversos hiperparâmetros de difícil ajuste.

O trabalho de MAHIMKAR *et al.* [44] propõe a ferramenta Giza para diagnosticar problemas em redes IPTV. O método utiliza como entrada a informação de chamados ao *call center*, o resultado de medições de diversas métricas de redes IPTV e os padrões de uso de usuários para a criação de grafos de dependências a partir dos quais se torna possível inferir a relação entre problemas detectados e suas possíveis causas. Limitações do trabalho incluem a ausência de uma análise que considere o ruído associado a chamados ao *call center* e a dificuldade associada à manutenção do grafo de dependências.

Métodos baseados em grafos de dependência compartilham certas limitações. A construção de grafos de dependência pode não ser simples e os grafos são quase sempre dinâmicos [43]. É essencial manter uma representação precisa da rede, uma vez que diferenças entre o grafo e a realidade podem afetar a eficácia da localização [43], mas o comportamento dinâmico e o tamanho dos grafos tornam esta manutenção desafiadora. Além disso, o custo computacional destas abordagens pode ser alto, o que obriga os métodos a considerarem heurísticas gulosas ou suposições como um número máximo de falhas.

2.5 Aprendizado de máquina

Abordagens baseadas na extração automática de padrões em resultados de medições de redes residenciais podem ser encontradas na literatura. De maneira similar ao nosso trabalho, métodos considerados nesta categoria têm como objetivo diagnosticar, detectar e/ou localizar falhas a partir de métodos de aprendizado aplicados a um dataset de medições realizadas em provedores de acesso.

Os autores de [45] propõem o método NEVERMIND com o objetivo de diagnosticar e solucionar problemas em redes de acesso DSL. Nesta abordagem, um classificador supervisionado Adaboost é treinado utilizando a existência de chamados como *labels* e medições esparsas de métricas de rede física DSL como *features* para prever chamados futuros a partir de medições. Para a localização de problemas utiliza-se um modelo *ensemble* baseado na estrutura hierárquica de redes DSL. O trabalho não leva em consideração o ruído associado a chamados de maneira apropriada, utilizando métodos supervisionados para o treinamento dos modelos. Além disso, o método é baseado em métricas específicas de redes DSL.

O framework Q-score é proposto por SONG *et al.* [46] com o objetivo de detectar problemas de rede que afetem a qualidade de experiência, medida a partir de chamados ao *call center*. Com este intuito, o método propõe classificadores supervisionados baseados em regressão de Ridge que utilizam chamados como *labels* e métricas extraídas a partir de medições de rede IPTV e atividade de usuários como *features*. A utilização de modelos supervisionados baseados em chamados ao *call center* não é adequada por conta do ruído associado ao comportamento dos usuários.

O sistema distribuído de classificação de falhas ICCA (*Intelligent Customer Care Assistant*) é proposto em [47] para identificar se um chamado recebido por uma operadora de rede móvel tem como causa uma falha na rede ou um problema no dispositivo do usuário. O método realiza o treinamento de um classificador baseado em *Gradient Boosted Decision Trees*, utilizando métricas de redes móveis e tráfego

do usuário como *features* e a classificação do tipo de chamado como *labels*. Este trabalho não propõe um método para localização de falhas e é baseado em *features* específicas de redes móveis, o que limita a sua aplicabilidade.

O sistema AESOP (*Automatic Extraction of Signatures for Operation Policies*) é proposto por DEB *et al.* [48] com o objetivo de aprender automaticamente políticas para mitigação de eventos que causam degradação no desempenho da rede. Para aprender quais ações corretivas devem ser aplicadas em cada cenário, o método realiza o *fit* de uma regressão logística utilizando como *features* os resultados de medições passivas em redes móveis e como *target* o histórico de ações corretivas realizadas por operadores de rede. O trabalho aplica diversas etapas de pré-processamento heurísticas, o que prejudica a capacidade de generalização do método para contextos não relacionados a redes móveis.

O trabalho de HU *et al.* [49] propõe o método CableMon para inferir falhas a partir de métricas de diagnóstico de redes HFC. Com o objetivo de detectar falhas que impactam a experiência do usuário, os autores propõem um classificador baseado em *thresholds* que utiliza como *features* métricas extraídas a partir de medições de rede física HFC e como *target* uma métrica relacionada à quantidade de chamados ao *call center*. O trabalho não propõe um método para localização de falhas e é baseado em métricas específicas de redes HFC.

Propomos neste trabalho um modelo de detecção de degradação de desempenho de rede baseado no conceito de rótulos com ruído. Um survey sobre modelos de classificação capazes de lidar com *noisy labels* pode ser encontrado em [50]. Modelos baseados em rótulos com ruído foram previamente aplicados em outras áreas de conhecimento, como visão computacional [51] e biomedicina [52]. ZHAO *et al.* [18] propõem o framework RAD para detectar anomalias de rede quando rótulos com ruído estão presentes. No entanto, a abordagem exige um dataset sem ruído de tamanho relevante para o treinamento do modelo inicial, um requisito custoso que pode ser de difícil obtenção. Em contraste, o modelo proposto neste trabalho é estimado utilizando apenas rótulos com ruído.

Uma limitação comum associada a métodos analisados nesta seção é relacionada ao *ground truth* utilizado para o *fit* de modelos. Métodos da literatura consideram como *labels* para o treinamento de classificadores chamados realizados ao *call center*, que possuem ruído associado ao comportamento dos usuários, ou um conjunto incompleto de rótulos obtidos através de inspeção manual, uma abordagem custosa e sujeita a erros. Desta forma, os rótulos considerados para a criação dos modelos da literatura possuem ruído associado. No entanto, poucos métodos considerados lidam com o ruído das *labels* de maneira apropriada, uma vez que métodos supervisionados tradicionais ou heurísticas são considerados para o treinamento dos modelos apesar do ruído existente. Além disso, os modelos são comumente treinados utilizando

dados relacionados a uma tecnologia de acesso específica, como DSL, HFC, IPTV ou tecnologias de acesso móvel, e classificadores são criados a partir de datasets com centenas de *features*. Consequentemente, os métodos propostos possuem aplicação associada a uma tecnologia de acesso específica e tendem a apresentar baixa generalização.

2.6 Contribuições de nossa proposta

A partir da análise dos trabalhos relacionados presentes na literatura, é possível identificar uma série de limitações que endereçamos nas técnicas propostas neste trabalho. Especificamente, nosso trabalho provê de maneira simultânea as seguintes características:

- Propomos métodos de detecção e localização de falhas a partir da extração automática de padrões de um conjunto de dados real de medições coletadas em um provedor de acesso residencial (diferentemente da maioria dos métodos da Seção 2.3). Consideramos um conjunto de dados com as seguintes características:
  - Medições ativas de rede com granularidade fina realizadas de maneira contínua a partir de *gateways* residenciais combinadas com informações de topologia e de chamados ao *call center* (diferentemente dos métodos da Seção 2.1).
  - Métricas extraídas a partir de medições que podem ser realizadas em qualquer tipo de rede, sem a utilização de indicadores específicos de uma determinada tecnologia de acesso (diferentemente da maioria dos métodos da Seção 2.5).
- Nosso método tem como foco identificar degradações de desempenho em redes de acesso residenciais (diferentemente dos métodos da Seção 2.2). Além disso, o método proposto neste trabalho é capaz de detectar falhas transientes, como congestionamentos de rede, e falhas persistentes, como falhas em equipamentos (diferentemente da maioria dos métodos da Seção 2.3).
- Todos os métodos propostos levam em consideração a correlação temporal entre as amostras e a possibilidade de variação na distribuição das métricas de rede em diferentes instantes de tempo, diferentemente da maioria dos métodos da literatura.

- É nosso objetivo identificar o estado da rede sem a utilização de limiares fixos (diferentemente da maioria dos métodos de tomografia binária da Seção 2.3). Especificamente, consideramos métodos não supervisionados de aprendizado de máquina que possibilitam a definição de uma qualidade aceitável de rede a partir da interpretação dos padrões detectados nos dados. Além disso, propomos um método de aprendizado de máquina baseado em rótulos com ruído que identifica automaticamente a partir dos dados os padrões que representam uma rede de boa (ou má) qualidade.

- Propomos um método que é capaz de inferir a qualidade latente da rede a partir de medições de qualidade do serviço e de rótulos com ruído, como chamados ao *call center* (Capítulo 6). Consideramos um método de otimização probabilístico que lida com o ruído de maneira explícita, sem a necessidade de um conjunto de rótulos confiáveis, diferentemente dos métodos da Seção 2.5. Não é de nosso conhecimento nenhum outro método de otimização de modelos para detecção de falhas em redes que considere apenas rótulos com ruído.

- Nossos métodos são eficientes computacionalmente para que possam ser aplicados em redes residenciais reais (diferentemente dos métodos da Seção 2.4). Demonstramos a aplicabilidade dos nossos métodos através de uma prova de conceito que detecta e localiza eventos de rede em tempo real (Seção 7.1.2).

A Tabela 2.1 sumariza a comparação entre os trabalhos discutidos neste capítulo e as técnicas propostas nesta tese.

Tabela 2.1: Comparação entre os trabalhos relacionados e as técnicas propostas neste trabalho. ✓: aspecto contemplado; ∼: aspecto contemplado parcialmente; em branco: aspecto não contemplado ou não reportado.

	Medições ativas	Redes de acesso (ISP)	Independente da tecnologia de acesso	Localização	Correlação temporal	Sem limiar fixo sobre as métricas	Múltiplas falhas simultâneas	Falhas transientes e persistentes	Usa chamados ao *call center*	Modela ruído dos rótulos	Avaliação com dados reais
Medições passivas (Seção 2.1)											
Padmanabhan et al. [23]			✓	✓			✓				✓
CEM [24]		∼	✓	✓	∼	∼	✓	∼			✓
Argus [16]		∼	✓	✓	✓	∼	✓	∼			✓
Medições em datacenters (Seção 2.2)											
Herodotou et al. [22]	✓		✓	✓	∼	✓	∼	∼			✓
Themis [27]	✓		✓	✓	∼	✓					∼
Pingmesh [29]	✓		✓					∼			✓
deTector [30]	✓		✓	✓			✓	∼			∼
Simon [32]	✓		✓	✓		✓	✓				∼
Zoom2Net [33]			✓		✓	✓					✓
μMon [34]			✓	∼	∼						∼
NetBouncer [28]	✓		✓	✓			✓				✓
Tomografia de rede (Seção 2.3)											
Cáceres et al. [35]	✓		✓	✓		✓	✓				
Presti et al. [36]	✓		✓	✓		✓	✓				
Duffield et al. [6]	✓		✓	✓		✓	✓				
NetScope [7]	✓		✓	✓		✓	∼				∼
Tomografia binária de rede (Seção 2.3)											
SCFS [5]	✓		✓	✓			∼				
NetDiagnoser [31]	✓		✓	✓			✓				
Nguyen e Thiran [37]	✓		✓	✓			✓				∼
Cunha et al. [38]	✓		✓	✓	∼		∼				∼
Barford et al. [17]	✓		✓	✓							
Ghita et al. [39]	✓		✓	∼			✓				
Hermes [40]	✓	∼	✓	✓	∼	✓	✓	∼			✓
Grafo de dependências (Seção 2.4)											
Shrink [41]			✓	✓			∼				
Sherlock [42]			✓	✓		✓	∼	✓			✓
Kompella et al. [43]	✓		✓	✓			✓				✓
Giza [44]		✓		✓	∼		✓	∼	✓		✓
Aprendizado de máquina (Seção 2.5)											
NEVERMIND [45]		✓		✓	∼	✓	✓	∼	✓		✓
Q-score [46]		✓		∼		✓	✓		✓		✓
ICCA [47]		✓			∼	✓			✓		✓
AESOP [48]		✓			∼	✓	✓				✓
CableMon [49]		✓			∼	∼	✓		✓		✓
RAD [18]			✓			✓				∼	∼
Este trabalho											
Decomposição tensorial (Seção 5.1)	✓	✓	✓	✓	✓	✓	✓	✓			✓
Pontos de mudança (Seção 5.2)	✓	✓	✓	✓	✓	✓	✓	✓			✓
Rótulos com ruído (Cap. 6)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Capítulo 3

Background

Descrevemos neste capítulo os conceitos utilizados como base para a construção dos métodos de detecção de eventos de rede propostos neste trabalho. Consideramos dois tipos de modelos de detecção de degradação de desempenho de rede para lidar com dois cenários distintos. No primeiro cenário, consideramos como entrada para o detector de degradação o resultado de medições de latência e perda, formando uma série temporal multivariada. Para este caso, consideramos um método baseado no modelo de decomposição tensorial PARAFAC, descrito na Seção 3.1. Para o segundo cenário, consideramos apenas os resultados das medições de perda de pacotes. Neste caso, consideramos uma série temporal de perda univariada modelada por Hidden Markov Models (HMMs), um modelo estocástico utilizado para descrever séries temporais descrito na Seção 3.2.

3.1 PARAFAC

Aplicamos neste trabalho métodos de análise de dados multivariados baseados no conceito de *tensores* para detectar degradações de desempenho em séries temporais multivariadas obtidas a partir de medições de latência e perda de milhares de residências distintas. Trabalhos anteriores da literatura consideram a aplicação de tensores para detecção de anomalias a partir de métricas relacionadas ao volume de tráfego, como o número de bytes/pacotes transmitidos entre pares Origem-Destino [8, 19–21], enquanto consideramos neste trabalho a aplicação de decomposição tensorial aos resultados de medições de latência e perda (Seção 5.1). Apresenta-se a seguir uma breve descrição do método de decomposição tensorial PARAFAC. Uma descrição mais detalhada pode ser encontrada em [53, 54].

Um tensor é uma generalização do conceito de matriz para um número qualquer de dimensões N. Na literatura de tensores, cada dimensão é denominada um *modo* do tensor, enquanto o número de dimensões é denominado a *ordem* de um tensor. Desta forma, vetores são tensores de primeira ordem, enquanto matrizes são tensores

de segunda ordem. Tensores com ordem maior ou igual a três são denominados tensores de alta ordem. Consideramos neste trabalho tensores de terceira ordem denotados por $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$.

Métodos de decomposição tensorial que generalizam o conceito de decomposição matricial podem ser utilizados para extrair de um tensor de dados padrões latentes que indicam eventos de interesse. Em particular, consideramos neste trabalho a utilização de decomposição tensorial para a extração de um subespaço normal com o objetivo de facilitar a detecção de degradação, uma técnica explorada na literatura de detecção de anomalias em diversas áreas, como mostra o *survey* de FANAEE-T e GAMA [55]. Aplicamos neste trabalho o método de decomposição tensorial de PARAFAC por conta de sua simplicidade, robustez e interpretabilidade [54]. Descrevemos a seguir o método de PARAFAC utilizando tensores de terceira ordem. No entanto, o método é geral e pode ser utilizado para tensores de alta ordem com um número qualquer de dimensões.

O método de PARAFAC realiza a aproximação de um tensor tridimensional $\mathcal{X}$ através da soma de R tensores de rank um representados utilizando as matrizes fatoriais $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ e $\mathbf{C} \in \mathbb{R}^{K \times R}$. Formalmente, o modelo é definido por:

$$\mathcal{X} = \mathcal{M} + \mathcal{E} \tag{3.1}$$

$$\mathcal{M}_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \tag{3.2}$$

Onde R representa o número de fatores utilizados na decomposição, $\mathcal{E}$ define o tensor residual e $\mathcal{M}$ representa o modelo obtido pelo método de PARAFAC.

O objetivo da decomposição de PARAFAC é encontrar o modelo $\mathcal{M}$ que minimiza a soma dos quadrados dos residuais para um determinado número de fatores R, ou seja, estimar as matrizes fatoriais $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ que minimizem o erro da aproximação obtida pelo modelo. Um algoritmo comumente utilizado na literatura para solucionar este problema é a técnica de *Alternating Least Squares* (ALS). A ideia deste método é otimizar de maneira alternada uma das matrizes fatoriais mantendo as outras matrizes fixas, o que faz com que cada passo do algoritmo seja reduzido ao cálculo dos mínimos quadrados.

O primeiro passo do algoritmo ALS consiste na atribuição de valores iniciais para as matrizes fatoriais, o que pode ser feito de maneira aleatória ou através de métodos matriciais como Singular Value Decomposition (SVD). Em seguida, otimiza-se alternadamente cada matriz fatorial de maneira iterativa até que um determinado critério de parada seja alcançado. Critérios de parada para o algoritmo incluem um limiar mínimo para a variação do erro, um limiar mínimo para a variação

das matrizes fatoriais e um número máximo de iterações. Considerando um tensor de terceira ordem, a otimização das matrizes fatoriais aplicada em cada iteração é obtida a partir das seguintes equações:

$$\mathbf{A} \leftarrow \mathcal{X}_{(1)}((\mathbf{C} \odot \mathbf{B})^T)^\dagger \tag{3.3}$$

$$\mathbf{B} \leftarrow \mathcal{X}_{(2)}((\mathbf{C} \odot \mathbf{A})^T)^\dagger \tag{3.4}$$

$$\mathbf{C} \leftarrow \mathcal{X}_{(3)}((\mathbf{B} \odot \mathbf{A})^T)^\dagger \tag{3.5}$$

Onde $\mathcal{X}_{(i)}$ representa a matricização do tensor $\mathcal{X}$ no modo i, o símbolo $\odot$ representa o produto de Khatri-Rao e o símbolo † representa a pseudo-inversa de Moore-Penrose da matriz. Não há garantias teóricas de que o método de ALS alcance um mínimo global do erro, uma vez que o modelo final obtido por esta técnica depende dos valores iniciais atribuídos às matrizes fatoriais.

O número de fatores R é um hiperparâmetro do método ALS e deve ser definido antes da aplicação do algoritmo. Para definir o número de fatores, consideramos neste trabalho a aplicação da técnica de Split-Half Validation (SV) [56]. Neste método, divide-se o *dataset* em 4 subconjuntos distintos (S_1, S_2, S_3, S_4) e avalia-se a similaridade entre as decomposições de PARAFAC obtidas pelos conjuntos $S_1 \cup S_2$ e $S_3 \cup S_4$, além da similaridade entre os modelos obtidos pelos conjuntos $S_1 \cup S_3$ e $S_2 \cup S_4$. O objetivo da técnica é avaliar se modelos obtidos com um determinado número de fatores R são generalizáveis. Espera-se que um modelo com um número inadequado de fatores R não seja robusto, ou seja, que modelos consideravelmente distintos sejam obtidos para diferentes subconjuntos de dados, indicando a possibilidade de *overfitting*. Nesta definição, consideramos que o modo $\mathbf{A}$ representa o modo das amostras e que o objetivo da validação é medir a capacidade de generalização do modelo para novas amostras, o que representa um dos testes mais comuns. Desta forma, a divisão dos conjuntos utilizados pelo SV é feita a partir do modo $\mathbf{A}$ e a análise de similaridade não inclui este modo, conforme previsto em [56].

Para avaliar a similaridade entre os modelos obtidos em cada um dos subconjuntos, utiliza-se o Tucker Congruence Coefficient (TCC) [57]. A partir das matrizes fatoriais $\mathbf{B}^{f_1}, \mathbf{C}^{f_1}$ calculadas a partir do primeiro conjunto de dados e das matrizes $\mathbf{B}^{f_2}, \mathbf{C}^{f_2}$ obtidas com a decomposição tensorial do segundo conjunto, o TCC é definido para cada fator r através das fórmulas:

$$\phi(\mathbf{b}_r^{f_1}, \mathbf{b}_r^{f_2}) = \frac{\sum_{j=1}^{J} b_{jr}^{f_1} b_{jr}^{f_2}}{\sqrt{\sum_{j=1}^{J}(b_{jr}^{f_1})^2 \sum_{j=1}^{J}(b_{jr}^{f_2})^2}} \tag{3.6}$$

$$\phi(\mathbf{c}_r^{f_1}, \mathbf{c}_r^{f_2}) = \frac{\sum_{k=1}^{K} c_{kr}^{f_1} c_{kr}^{f_2}}{\sqrt{\sum_{k=1}^{K}(c_{kr}^{f_1})^2 \sum_{k=1}^{K}(c_{kr}^{f_2})^2}} \tag{3.7}$$

O número de fatores R é validado quando se obtêm valores para o TCC maiores que 0.95 para cada fator em cada divisão considerada pelo SV, conforme recomendado em [57].

3.2 Hidden Markov Models

Para realizar a modelagem de séries temporais univariadas de perda, consideramos a utilização de Hidden Markov Models (HMMs). No contexto de redes de computadores, trabalhos anteriores consideraram modelos baseados em HMMs para aplicações como a modelagem de um canal de comunicação [58], a criação de modelos para simulação e emulação do comportamento da rede [59] e a otimização de algoritmos de *Forward Error Correction* (FEC) [60]. Consideramos neste trabalho a utilização de Hidden Markov Models para identificar degradações de desempenho a partir da detecção de pontos de mudança estatística na distribuição das séries temporais de perda (Seção 5.2) e para a detecção de degradação utilizando rótulos com ruído (Capítulo 6). Apresentamos a seguir uma breve descrição do funcionamento de HMMs. Detalhes sobre a implementação de Hidden Markov Models podem ser encontrados em [61, 62].

Uma Hidden Markov Model é um processo estocástico utilizado para modelar séries temporais, denotadas por $\mathbf{O} = (O_1, O_2, ..., O_T)$, onde T representa a quantidade de amostras da série. Uma HMM associa a cada amostra da série temporal O_t, denominada símbolo observável, uma variável latente s_t, denominada estado oculto. A distribuição de probabilidade de símbolos observáveis é completamente definida dado o estado oculto atual, ou seja, $P(O_t|s_t)$ é conhecida. Além disso, a transição entre estados ocultos possui a propriedade de Markov, ou seja, $P(s_t|s_{t-1}, s_{t-2}, ..., s_1) = P(s_t|s_{t-1})$. A Figura 3.1 mostra a representação de uma HMM utilizando um Probabilistic Graphical Model (PGM).

Diversas distribuições discretas ou contínuas podem ser utilizadas para a modelagem dos símbolos observáveis. Consideramos neste trabalho HMMs em que os símbolos observáveis seguem uma distribuição categórica. Neste cenário, uma cadeia oculta é definida através de 2 hiperparâmetros: o número de estados ocultos N e o número de símbolos observáveis distintos M. Além disso, são parâmetros da cadeia

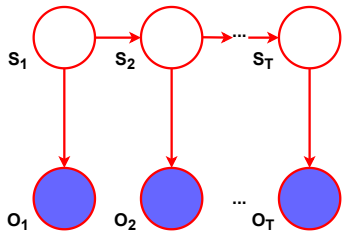


Figura 3.1: Hidden Markov Models representadas através de um Probabilistic Graphical Model

de Markov oculta o vetor com a distribuição inicial de estados ocultos $\Pi \in \mathbb{R}^N$, a matriz de transição entre estados ocultos $\mathbf{A} \in \mathbb{R}^{N \times N}$ e a matriz de observações $\mathbf{B} \in \mathbb{R}^{N \times M}$, cujos componentes são descritos através das seguintes equações:

$$\pi_j = P(s_1 = j) \tag{3.8}$$

$$a_{jk} = P(s_t = k | s_{t-1} = j) \tag{3.9}$$

$$b_{jl} = P(O_t = l | s_t = j) \tag{3.10}$$

O conjunto de parâmetros de uma HMM pode ser descrito utilizando a notação compacta $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$.

Para a aplicação de HMMs em cenários reais é necessária a resolução de três problemas de interesse descritos em RABINER [61]:

1. Calcular a probabilidade do modelo gerar uma determinada sequência de observações $P(\mathbf{O})$

2. Obter a partir do modelo uma sequência ótima de estados ocultos $\mathbf{S} = (s_1, s_2, ..., s_T)$ dada uma sequência de observações $\mathbf{O} = (O_1, O_2, ..., O_T)$

3. Otimizar os parâmetros do modelo $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$

A solução dos problemas enumerados é baseada no cálculo de um conjunto de variáveis auxiliares definidas por:

$$\alpha_t(j) = P[O_1, O_2, ..., O_t, s_t = j] \tag{3.11}$$

$$\beta_t(j) = P[O_{t+1}, O_{t+2}, ..., O_T | s_t = j] \tag{3.12}$$

$$\gamma_t(j) = P[s_t = j | \mathbf{O}] = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{3.13}$$

$$\begin{aligned} \xi_t(j,k) &= P[s_t = j, s_{t+1} = k | \mathbf{O}] \\ &= \frac{\alpha_t(j) a_{jk} b_{k(O_{t+1})} \beta_{t+1}(k)}{\sum_{i=1}^{N} \sum_{m=1}^{N} \alpha_t(i) a_{im} b_{m(O_{t+1})} \beta_{t+1}(m)} \end{aligned} \tag{3.14}$$

O primeiro problema pode ser resolvido através do algoritmo de programação dinâmica *Forward*, utilizado para calcular o valor das variáveis auxiliares $\alpha_t(j)$ [61]. Após a aplicação do algoritmo *Forward*, a probabilidade de uma sequência de observações $\mathbf{O}$ é calculada por:

$$P(\mathbf{O}) = \sum_{j=1}^{N} \alpha_T(j) \tag{3.15}$$

A solução do segundo problema depende da definição de otimalidade para uma sequência de estados ocultos. Quando o objetivo é maximizar o número esperado de estados individualmente corretos, a solução ótima é obtida utilizando a variável auxiliar $\gamma_t(j)$ através da equação:

$$q_t = \underset{1 \leq j \leq N}{\operatorname{argmax}}[\gamma_t(j)] \tag{3.16}$$

Quando o objetivo é encontrar a sequência de estados ocultos mais provável dada a sequência de observações, utiliza-se o método de Viterbi [61], um algoritmo de programação dinâmica que encontra de maneira eficiente a sequência de estados que maximiza $P(\mathbf{S}|\mathbf{O})$.

Por fim, a solução do terceiro problema é dada a partir do método de Baum-Welch [61], um algoritmo iterativo que obtém uma estimativa de máxima verossimilhança (MLE) para os parâmetros do modelo a partir de uma sequência de observações. O algoritmo de Baum-Welch pode ser visto como um caso particular do método de Expectation-Maximization (EM), onde as variáveis auxiliares definidas nas equações 3.11, 3.12, 3.13 e 3.14 são avaliadas durante o passo E enquanto os parâmetros do modelo $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$ são maximizados durante o passo M. É possível demonstrar que as equações utilizadas para maximizar os parâmetros do modelo durante o passo M são dadas por:

$$\pi_i = \gamma_1(i) \tag{3.17}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.18}$$

$$b_{ik} = \frac{\sum_{t=1}^{T} \gamma_t(i) \mathbb{1}(O_t = k)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.19}$$

Onde $\mathbb{1}(O_t = k)$ é uma variável aleatória indicadora que vale 1 quando $O_t = k$ e 0 caso contrário.

Capítulo 4

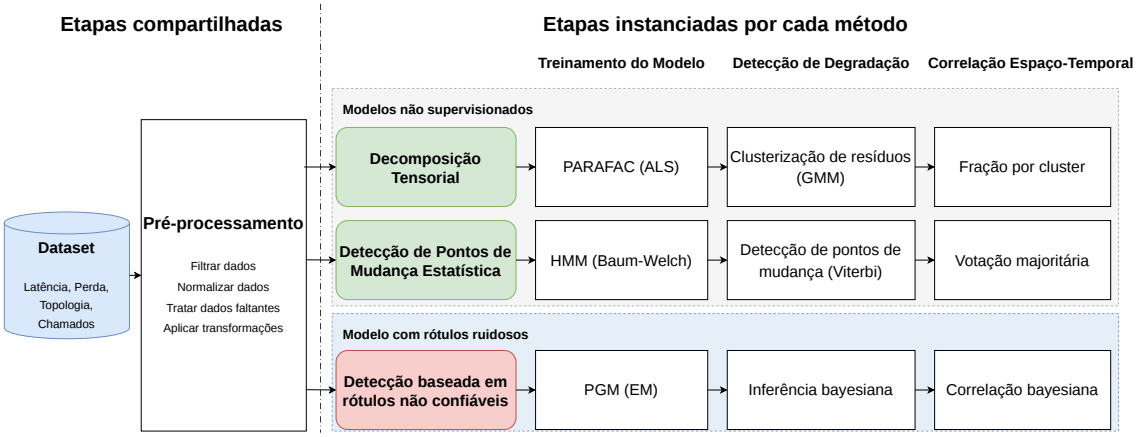
Metodologia

Apresentamos neste capítulo a metodologia desenvolvida neste trabalho para a detecção e localização de degradação no desempenho da rede. Todas as técnicas elaboradas nesta tese adotam um mesmo framework para detectar e localizar eventos que causam degradação na qualidade do serviço oferecido pelo ISP. Os passos adotados pelo framework proposto são descritos na Seção 4.1. As técnicas construídas se baseiam na extração de padrões presentes em dados reais através da utilização de aprendizado de máquina. A construção do dataset, obtido a partir de medições de QoS, dados sobre a topologia da rede e chamados realizados ao *call center* do ISP, é detalhada na Seção 4.2.

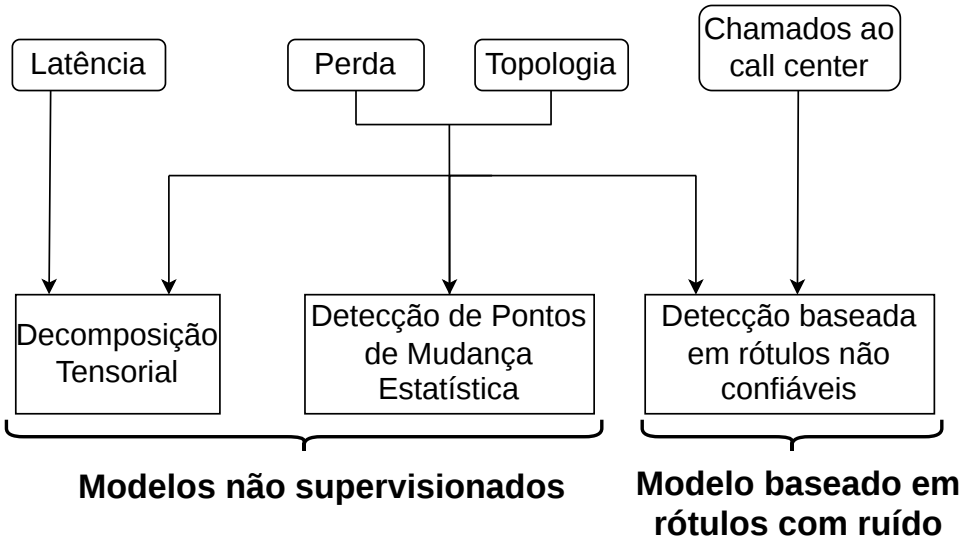
4.1 Framework

A Figura 4.1a apresenta as etapas do framework construído para detectar e localizar degradações no desempenho da rede por meio de técnicas de aprendizado de máquina. Para o cenário em que não temos rótulos, consideramos dois métodos não supervisionados distintos: um método baseado em decomposição tensorial, capaz de extrair padrões a partir de séries temporais multivariadas de latência e perda, e um método baseado na detecção de pontos de mudança estatística, capaz de identificar degradação de maneira online utilizando séries temporais de perda. Para o cenário em que dados de chamados ao call center estão disponíveis, consideramos um novo método para detecção baseada em rótulos com ruído utilizando séries temporais de perda. Cada etapa do framework varia de acordo com o método considerado. Detalhamos a seguir os passos adotados em nosso framework:

- **Pré-processamento**: Etapa que inclui a limpeza e filtragem dos dados, a aplicação de técnicas para lidar com dados faltantes e a aplicação das demais transformações necessárias para o treinamento do modelo. O tratamento dos dados varia de acordo com o modelo a ser aplicado. Por exemplo: o método



(a) Framework para detecção e localização de degradação de desempenho em redes



(b) Técnicas propostas para detectar e localizar degradação de desempenho da rede

Figura 4.1: Visão geral da abordagem proposta

de decomposição tensorial considera a aplicação da escala logarítmica para normalização, enquanto as técnicas baseadas em Cadeia de Markov Oculta utilizam um conjunto de símbolos observáveis extraídos a partir das medições de QoS.

- **Treinamento do modelo**: Nesta etapa, realiza-se o treinamento do modelo de aprendizado de máquina adotado para detectar degradação de desempenho, o qual varia de acordo com o método considerado. Esta etapa também inclui a definição dos hiperparâmetros necessários para a aplicação do algoritmo de treinamento. Por conta da escassez de rótulos confiáveis, consideramos em nosso trabalho dois tipos diferentes de modelos: modelos não supervisionados (Capítulo 5) para cenários em que não existem rótulos e modelos baseados em rótulos com ruído (Capítulo 6) para cenários em que temos acesso a uma base de dados de chamados ao call center.

- **Detecção de degradação na rede**: Neste passo são implementadas as técnicas de detecção de degradação do desempenho da rede, construídas a partir dos resultados obtidos pela aplicação dos modelos treinados aos dados pré-processados de cada residência. O objetivo desta etapa é identificar de maneira automática quais residências experimentam degradação na qualidade do serviço oferecido pelo ISP. A técnica utilizada varia de acordo com o modelo considerado: adotamos a clusterização dos resíduos para o método baseado em decomposição tensorial, a identificação de transições entre estados ocultos para a técnica baseada em detecção de pontos de mudança e a estimativa da probabilidade de degradação da rede para a abordagem baseada em rótulos com ruído.

- **Correlação Espaço-Temporal**: A última etapa do framework correlaciona espaço-temporalmente os resultados das detecções de degradação de desempenho realizadas em diferentes residências para determinar quais regiões da rede experimentam degradação de desempenho. Combinamos os resultados da etapa anterior com o detalhamento da topologia disponibilizado pelo ISP para localizar regiões potencialmente afetadas pela degradação identificada pelo nosso método.

Aplicamos o framework acima para construir três abordagens de detecção e localização de degradação de desempenho apresentadas na Figura 4.1b. Consideramos inicialmente técnicas não supervisionadas por conta da dificuldade de obtenção de rótulos (Capítulo 5). Apresentamos na Seção 5.1 uma abordagem baseada na decomposição tensorial de séries temporais multivariadas de latência e perda que utiliza a informação da topologia para detectar e localizar degradação de maneira não supervisionada. Trabalhos anteriores mostram que a perda de pacotes tem maior impacto na QoE dos usuários quando comparada à latência [63, 64]. Além disso, é de nosso interesse a criação de métodos capazes de detectar degradação de desempenho de maneira *online* para reduzir o tempo de resposta do ISP a problemas detectados. Desta forma, descrevemos na Seção 5.2 uma segunda técnica não supervisionada baseada na detecção de pontos de mudança estatística nas séries temporais de perda que utiliza a informação da topologia para detectar e localizar degradação de desempenho de maneira *online*. Por fim, consideramos o cenário em que rótulos com ruído estão disponíveis para o treinamento de modelos. Propomos uma nova técnica para detectar e localizar degradação de desempenho em redes utilizando séries temporais de QoS e rótulos de QoE com ruído, apresentada no Capítulo 6. A aplicação desta abordagem é exemplificada utilizando os resultados de medições de perda como métrica de QoS e os chamados ao call center como rótulos de QoE com ruído.

4.2 Construção do Dataset

Descrevemos nesta seção a metodologia aplicada para construir o conjunto de dados utilizado para a aplicação dos métodos propostos. Todos os dados considerados foram obtidos a partir de uma parceria entre a universidade, uma *startup* iniciada dentro da incubadora da universidade e um ISP de médio porte. A infraestrutura de medição é detalhada na Seção 4.2.1, enquanto as informações disponibilizadas pelo ISP são descritas na Seção 4.2.2.

4.2.1 Infraestrutura de Medição

Figura 4.2: Infraestrutura de medição

A Figura 4.2 mostra a arquitetura do nosso ambiente de medição. As medições de desempenho da rede consideradas neste trabalho são obtidas a partir de roteadores residenciais com sistema operacional OpenWRT [1]. Estes roteadores são conectados diretamente à rede de acesso do ISP, funcionando como um *gateway* entre a rede residencial e a rede de acesso do provedor. A realização de medições periódicas a partir do *gateway* residencial fornece uma visão mais precisa da qualidade do serviço prestado ao usuário em contraste com a coleta realizada no interior da rede do ISP [25]. Consideramos neste trabalho as métricas de latência bidirecional, perda bidirecional, tráfego *upstream* e tráfego *downstream*.

As métricas de latência e perda são obtidas a partir de medições ativas realizadas através do envio de um trem de 100 pacotes ICMP, destinados a um servidor de medição dentro da rede do ISP. O intervalo entre pacotes de um mesmo trem é de 10 milissegundos, de forma a capturar períodos curtos de congestionamento sem onerar os recursos da rede em links de menor capacidade. Ressaltamos que o ISP não aplica filtragem nem priorização de pacotes ICMP. Além disso, as rotas entre os roteadores residenciais e o servidor de medição incluem apenas equipamentos dentro

[1] www.openwrt.org

da rede do ISP. Consideramos para análise a média da latência obtida pelos pacotes recebidos com sucesso e o número total de pacotes perdidos no trem de 100 pacotes.

As medições de tráfego *upstream* e *downstream* são realizadas de maneira passiva. Consideramos o tráfego em bits por segundo medido através de contadores de bytes enviados e recebidos pela interface WAN do roteador. O tráfego residencial pode afetar o resultado das medições de desempenho de rede [25]. Desta forma, utilizamos o resultado das medições de tráfego para filtrar amostras de latência e perda cujos resultados possam ter sido afetados. É importante frisar que medimos apenas o tráfego agregado utilizando contadores de tráfego nas interfaces de rede. Consequentemente, não temos acesso a informações sobre cabeçalhos de pacotes enviados ou recebidos por usuários do ISP, ou seja, a privacidade dos usuários é preservada.

O intervalo entre medições no nosso dataset é de um minuto, uma granularidade mais fina do que a considerada pela grande maioria dos trabalhos da literatura. Desta forma, para cada métrica do nosso dataset obtemos no máximo 1440 amostras diárias. As métricas de desempenho consideradas apresentam padrões diários relacionados ao comportamento dos usuários. Por exemplo, medições de latência e perda costumam apresentar valores mais baixos durante o período da madrugada, quando poucos usuários estão ativos e o tráfego na rede é baixo, e valores mais altos durante a noite, quando o tráfego na rede alcança o seu pico. Desta forma, é natural considerar para a análise as séries temporais diárias de cada métrica de interesse obtidas nas residências com capacidade de medição. Portanto, as análises realizadas nesta tese são baseadas em séries temporais diárias independentes obtidas em diferentes residências, denominadas *pares Residência-Dia (RD)*.

4.2.2 Dados disponibilizados pelo ISP

A parceria com o ISP possibilitou o acesso a informações sobre a topologia da rede e aos chamados realizados pelos usuários. A Figura 4.3 provê uma descrição em alto nível da topologia da rede em que as medições são realizadas. Consideramos uma divisão da rede em diferentes subconjuntos sem sobreposição, denominados *regiões da rede*. A divisão em regiões é baseada em um particionamento geográfico da cidade realizado pelo provedor de serviço. Cada região contém equipamentos de rede próximos geograficamente cuja responsabilidade consiste em conectar as residências da região ao *backbone* da rede. A infraestrutura de acesso de cada região pode ser representada a partir de uma árvore em que cada nó representa um equipamento de rede. O ISP disponibiliza informações sobre os equipamentos de cada árvore e suas interconexões. Além disso, também é informado a qual nó cada roteador de medição está conectado. Estas informações possibilitam a correlação espacial dos resultados das medições.

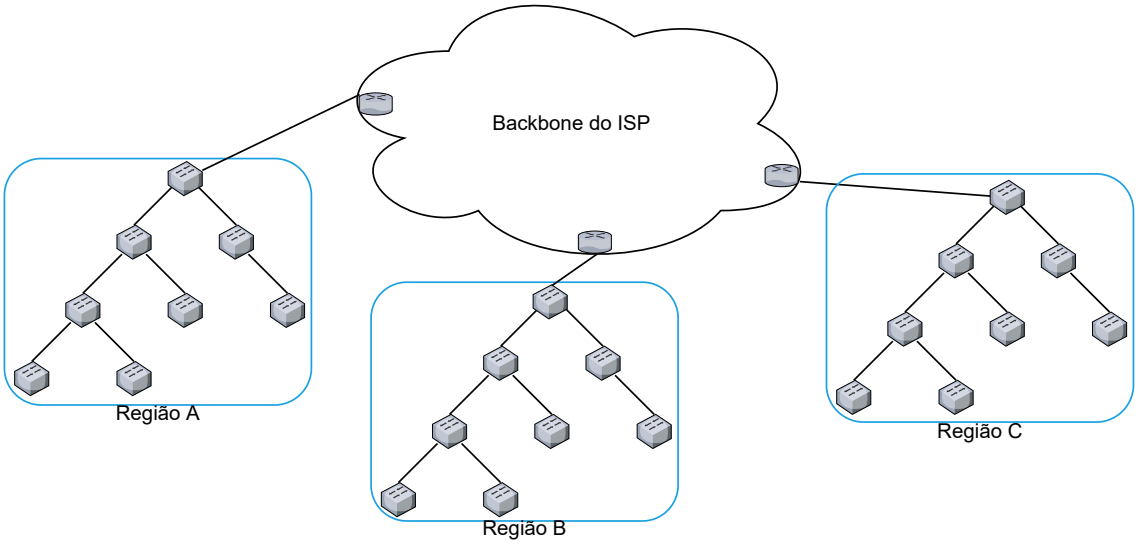


Figura 4.3: Partição da rede em regiões

O ISP também disponibiliza acesso a uma base de dados com informações sobre chamados técnicos realizados ao *call center*. A partir desta base, é possível identificar com granularidade diária em qual nó da topologia cada residência associada a um chamado está conectada. Dados de chamados não são indicadores confiáveis para a existência de falhas na rede, uma vez que usuários podem ou não criar chamados quando um problema de rede ocorre e vice-versa [49]. Desta forma, propomos nesta tese uma nova técnica capaz de lidar com o ruído associado aos chamados para detectar e localizar degradações de desempenho. É importante frisar que não utilizamos dados pessoais dos usuários durante a análise dos chamados: nosso banco de dados associa um ID anônimo de usuário ao identificador do roteador de medição, ou seja, a privacidade dos usuários é preservada.

4.3 Resumo

Apresentamos neste capítulo o framework proposto para detecção e localização de degradação de desempenho em redes residenciais. Um dos principais motivadores das técnicas propostas neste trabalho é a escassez de rótulos identificando degradações de desempenho, um problema comumente encontrado na literatura. Desta forma, dividimos os métodos propostos em duas categorias: *métodos não supervisionados*, onde consideramos modelos para a detecção de problemas na rede a partir de medições de QoS sem a existência de rótulos, e *métodos baseados em rótulos com ruído*, onde consideramos detectores que levam em consideração durante o processo de otimização medições de QoS e rótulos com ruído.

Extraímos a partir de medições realizadas continuamente um conjunto de dados real para a criação dos modelos de aprendizado de máquina. Especificamente, consideramos séries temporais de latência e/ou perda, extraídas a partir de medições realizadas em roteadores residenciais, como as métricas de QoS utilizadas para a detecção de degradação. Consideramos a existência de chamados ao *call center* como rótulos com ruído utilizados para auxiliar a identificação de problemas na rede. Também utilizamos a informação da topologia da rede do ISP para propor métodos capazes de localizar equipamentos experimentando degradação de desempenho através da correlação espacial dos resultados das medições.

Detalhamos nos próximos capítulos as técnicas construídas utilizando o framework apresentado neste capítulo. As abordagens de detecção e localização de degradação não supervisionadas são descritas no Capítulo 5, enquanto a nova técnica baseada em medições de QoS e rótulos de QoE com ruído é apresentada no Capítulo 6.

Capítulo 5

Métodos Não Supervisionados

Este capítulo detalha os métodos não supervisionados propostos para detecção de degradação de desempenho de rede. A Seção 5.1 apresenta uma abordagem que utiliza os resíduos obtidos pela decomposição tensorial de PARAFAC para detectar degradação no desempenho da rede de maneira não supervisionada a partir de séries temporais multivariadas extraídas dos resultados de medições de latência e perda. A aplicação do método multivariado proposto ao nosso dataset mostrou que a perda de pacotes apresenta maior impacto na qualidade da experiência dos usuários quando comparada à latência dos clientes avaliados (Seção 7.1.1). Desta forma, propomos um método de detecção de pontos de mudança na distribuição estatística de séries temporais de perda baseado em Hidden Markov Models com o objetivo de localizar regiões da rede que experimentam degradação de desempenho de maneira *online*, descrito na Seção 5.2.

5.1 Método baseado em Decomposição Tensorial

Descrevemos nesta seção um método capaz de detectar degradações de desempenho de maneira não supervisionada a partir dos resultados de medições de múltiplas métricas de rede formando um conjunto de séries temporais multivariadas. Exemplificamos a aplicação do método utilizando os resultados de medições contínuas de latência e perda realizadas por milhares de roteadores espalhados pela rede do ISP parceiro.

O método proposto utiliza um tensor de terceira ordem para modelar séries temporais multivariadas de diferentes clientes. Outros trabalhos da literatura utilizam para detecção de anomalias os resíduos de métodos de decomposição matricial como o PCA [2, 15] aplicados a tensores. No entanto, a aplicação de um método matricial em um tensor de mais de 2 dimensões exige a transformação do tensor em uma matriz através do processo de matricização. Além disso, o método de PCA é mais sensível à dimensionalidade do subespaço normal [65]. Utilizamos no nosso trabalho

a técnica de PARAFAC para modelar o subespaço normal através da decomposição tensorial, um método mais robusto ao ruído e capaz de preservar a correlação entre as diferentes métricas de séries temporais multivariadas. Em seguida, extraímos os resíduos utilizando o modelo obtido e aplicamos um método de clusterização destes resíduos para identificar padrões de degradação de desempenho. Por fim, correlacionamos os resultados de múltiplos clientes para identificar regiões experimentando degradação de desempenho.

O método proposto foi publicado como parte de um trabalho conjunto baseado em decomposição tensorial [10–12]. Uma discussão sobre o custo computacional associado ao método de PARAFAC pode ser encontrada em [66]. Detalhamos a seguir cada uma das etapas do nosso método.

5.1.1 Pré-processamento

Nesta etapa são realizadas as transformações e filtragens necessárias para a aplicação do método de decomposição tensorial. Consideramos no nosso trabalho os resultados das medições de latência e perda realizadas por diferentes pares Residência-Dia (RD). Medições ativas de latência e perda podem ser afetadas pelo tráfego residencial [25]. Para evitar o ruído associado a fatores externos à rede do ISP, filtramos as amostras de minutos em que o tráfego residencial esteja acima de um limiar θ e as amostras em minutos de indisponibilidade do servidor de medição durante o período considerado. Como o método de PARAFAC é capaz de lidar com dados faltantes durante o processo de otimização, representamos amostras filtradas por tráfego residencial concorrente ou indisponibilidade do servidor como dados faltantes. Além disso, consideramos apenas séries temporais diárias com pelo menos ψ amostras.

Certos eventos de rede, como rompimento de fibras ou falhas em equipamentos, podem interromper a comunicação entre o cliente e o servidor de medição, gerando períodos em que nenhuma medição apresenta resultados. Desta forma, é possível inferir períodos de indisponibilidade da rede através da ausência de resultados de medição, em especial quando múltiplas residências deixam de reportar resultados de maneira simultânea. Para lidar com este cenário, codificamos minutos sem resultados de medição como uma amostra de indisponibilidade, representada por uma fração de perda igual a 1. Por fim, consideramos a escala logarítmica para a aplicação do método de decomposição tensorial.

5.1.2 Treinamento do modelo

Realizamos nesta etapa a extração do subespaço normal através da decomposição tensorial utilizando a técnica de PARAFAC aplicada a amostras obtidas após o pré-processamento. Métodos anteriores da literatura consideram para a aplicação do

método de PARAFAC um tensor de terceira ordem com dimensões (IP origem x IP destino x Timestamp ou Porta) extraído a partir do cabeçalho de pacotes [19]. O trabalho de STREIT *et al.* [67] propôs a utilização de um tensor de terceira ordem com dimensões (Par RD x métrica x minuto) considerando como métricas a quantidade de bytes enviados e recebidos. Neste trabalho também consideramos como entrada para o ajuste do modelo um tensor com dimensões (Par RD x métrica x minuto), mas aplicamos o método de decomposição tensorial aos resultados das medições de latência e perda. Ao final deste processo, extrai-se do tensor de dados $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ o modelo $\mathcal{M}$ definido a partir das matrizes fatoriais $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ e $\mathbf{C} \in \mathbb{R}^{K \times R}$. Para definir o número de fatores R do modelo, utilizamos o método de Split-Half Validation. Mais detalhes sobre o processo de decomposição tensorial podem ser encontrados na Seção 3.1.

5.1.3 Detecção de degradação

Após a etapa de decomposição tensorial, responsável pela extração do subespaço normal, realizamos a extração dos resíduos a partir dos quais detectamos degradações. Os resíduos são obtidos através da diferença entre o tensor formado pelos resultados de medição e o modelo extraído pela decomposição tensorial, isto é, $\mathcal{E} = \mathcal{X} - \mathcal{M}$.

Os resultados de [12] mostram que a detecção baseada nos resíduos da decomposição tensorial é mais eficaz do que a realizada a partir dos *loadings* do modelo, exemplificando como a distribuição dos resíduos pode ser visualmente distinta em um cenário em que rótulos estão disponíveis. Para identificar degradação de desempenho na ausência de rótulos, propomos a aplicação de um método de clusterização utilizando *features* extraídas a partir dos resíduos da decomposição tensorial, com o objetivo de identificar padrões associados à degradação do desempenho da rede. Neste cenário, a identificação da degradação de desempenho depende da interpretação dos clusters obtidos.

A clusterização é realizada a partir de métricas extraídas de três tipos de séries temporais residuais: resíduos de latência, resíduos de perda e resíduos de perda filtrando amostras com perda igual a 1. Cada métrica auxilia na detecção de um tipo diferente de evento de rede. Por exemplo, residuais de perda não filtrada podem identificar eventos como rompimento de cabo e falha em equipamentos, enquanto residuais de latência e perda filtrada podem detectar congestionamentos na rede. A partir de cada série temporal de residuais, extraímos três *features* estatísticas: média, desvio padrão e 95^o percentil, totalizando 9 *features*.

Para realizar a clusterização de residuais de maneira não supervisionada, consideramos o método de Mistura de Gaussianas devido à sua flexibilidade e interpretabilidade. Para a escolha do número de *clusters* consideramos a aplicação do método

do cotovelo (*Elbow Method*), uma abordagem utilizada em trabalhos anteriores [2]. Antes de aplicar o método de clusterização, aplicamos a normalização *z-score* para evitar que a escala das *features* consideradas influencie os clusters obtidos.

Os nossos resultados (Seção 7.1.1) permitem associar a cada cluster um nível diferente de degradação no desempenho da rede, identificando degradações que impactam a experiência dos usuários.

5.1.4 Correlação Espaço-Temporal

Na última etapa do framework proposto, realizamos a correlação espaço-temporal dos resultados de medições de clientes próximos geograficamente, combinando os resultados da clusterização de residuais com a informação de topologia da rede do ISP parceiro. A ideia por trás do método é que clientes que compartilham um mesmo conjunto de *links* tendem a observar um desempenho similar da rede, o que se traduz em múltiplos clientes próximos geograficamente associados a um mesmo cluster de residuais.

Para aplicar o método de correlação espacial, particionou-se a rede em subconjuntos distintos não sobrepostos denominados regiões da rede. Diferentes métodos podem ser considerados para realizar este particionamento. Para o nosso dataset, esta divisão é realizada de acordo com um particionamento geográfico das cidades realizado pelo ISP parceiro (Seção 4.2), onde cada região contém múltiplos equipamentos de rede compartilhados por múltiplos clientes. Analisamos para cada região da rede a fração de usuários associados a cada cluster em cada dia com o objetivo de identificar padrões associados à degradação de desempenho afetando múltiplos clientes próximos geograficamente de maneira simultânea. Os nossos resultados (Seção 7.1.1) mostram que o método de clusterização espaço-temporal é capaz de identificar diferentes eventos de rede, como ataques DDoS, falhas em equipamentos e o impacto da pandemia de COVID-19. Além disso, o método pode auxiliar o ISP a priorizar esforços do seu time de operação ao identificar regiões da rede com pior desempenho.

5.2 Método baseado em Detecção de Pontos de Mudança

Os resultados obtidos pelo método multivariado baseado em PARAFAC mostram que, para as residências do nosso conjunto de dados, a latência tem menos impacto na qualidade da experiência dos usuários do que a perda de pacotes (Seção 7.1.1). Outros trabalhos da literatura também mostram que a perda de pacotes tem maior impacto na QoE quando comparada à latência [63, 64]. Além disso, a aplicação

online do método de decomposição tensorial é dificultada pela ausência de rótulos, uma vez que os clusters obtidos mudariam ao longo do tempo e seria inviável interpretá-los continuamente. Motivados por essas observações, propomos técnicas de detecção de degradação baseadas em séries temporais univariadas e demonstramos sua aplicação aos resultados de medições de perda de pacotes.

Detalhamos nesta seção o método não supervisionado de detecção e localização de eventos de rede baseado na detecção de pontos de mudança realizada a partir de Hidden Markov Models. Utilizamos a interpretabilidade do modelo de HMMs em conjunto com um filtro recursivo para identificar em cada residência com um equipamento de medição quais são os períodos em que a rede apresenta pior desempenho. Em seguida, correlacionamos espaço-temporalmente os resultados obtidos por cada medidor para localizar automaticamente quais equipamentos estão sendo afetados por eventos de rede que degradam a qualidade do serviço e a qualidade de experiência dos usuários.

O framework proposto foi publicado em [13]. Uma discussão sobre o custo computacional dos algoritmos utilizados por Cadeias de Markov Ocultas pode ser encontrada em [61]. Detalhamos nas subseções seguintes o funcionamento de cada etapa do método.

5.2.1 Pré-processamento

O primeiro passo da aplicação do método é a filtragem dos dados de medição. Consideramos um dataset formado pelos resultados de medições de perda de pacotes realizadas em diferentes residências por vários dias, os quais formam pares Residência-Dia (RD) distintos. Para localizar eventos na rede do ISP afetando múltiplos usuários, é importante filtrar amostras cujo resultado possa ser afetado pelo tráfego na rede residencial, uma vez que o resultado de medições de perda pode ser afetado pelo tráfego dos usuários [25]. Desta forma, filtramos todos os minutos de medição com tráfego residencial acima de um limiar θ. Além disso, consideramos para a análise apenas séries temporais com um número mínimo de amostras ψ. As amostras filtradas são marginalizadas durante o período de otimização dos parâmetros do modelo de HMM [68]. Por fim, é essencial inferir períodos de indisponibilidade para a detecção de eventos de rede como rompimento de cabo e falhas em equipamentos. Desta forma, codificamos amostras faltantes utilizando um símbolo observável ϕ.

5.2.2 Treinamento do modelo

Para treinar o modelo de HMM utilizamos o algoritmo de Baum-Welch, um caso particular do método de Expectation-Maximization (EM) (Seção 3.2), com o objetivo de estimar os parâmetros do modelo $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$. Consideramos para a

modelagem das séries temporais univariadas de perda uma distribuição categórica para os símbolos observáveis.

Para realizar o ajuste do modelo é necessário escolher como hiperparâmetros a quantidade de estados ocultos N e a quantidade de símbolos observáveis M. Além disso, o algoritmo de Baum-Welch é iterativo e encontra um ótimo local para a *likelihood* dos dados utilizados no treinamento que depende dos valores iniciais utilizados para os parâmetros do modelo $\mathbf{A}$, $\mathbf{B}$ e Π. A escolha destes parâmetros foi baseada na comparação da *log-likelihood* obtida pelo modelo em diferentes condições iniciais.

5.2.3 Detecção de degradação

A abordagem proposta para detectar degradações é baseada na detecção de pontos de mudança na distribuição estatística de séries temporais. Formalmente, definimos uma série temporal $\mathbf{O} = (O_1, O_2, ..., O_T)$, onde O_t representa o valor da série temporal no tempo t e o número total de amostras é representado por T. O objetivo de uma técnica de detecção de ponto de mudança é identificar o número de pontos de mudança k e o conjunto de pontos $\tau = (\tau_1, ..., \tau_k)$ tal que o intervalo (τ_i, τ_{i+1}) possua características estatísticas distintas das observadas antes de τ_i e depois de τ_{i+1}.

Diferentes métodos de detecção de pontos de mudança podem ser encontrados na literatura [69, 70]. Mais recentemente, o trabalho de DE ALMEIDA *et al.* [71] propõe um método de detecção de pontos de mudança estatística para cenários em que rótulos estão disponíveis. Nossa abordagem é baseada na utilização de HMMs para a detecção de pontos de mudança, um método previamente utilizado para este fim em aplicações distintas [72–74] que apresenta como vantagens sua interpretabilidade, robustez a *outliers* e parametrização mais simples quando comparada a outras técnicas. Nossos resultados mostram que o uso de HMMs é eficiente no contexto de detecção de degradação de desempenho em redes a partir de séries temporais de perda quando rótulos não estão disponíveis (Seção 7.1.2). Além disso, mostramos como o método pode ser estendido para criar um modelo estatístico de detecção baseada em rótulos com ruído (Capítulo 6).

Para identificar pontos de mudança, partimos da suposição de que diferentes estados ocultos possuem diferentes distribuições de símbolos observáveis, o que costuma ocorrer quando o número de estados ocultos N é definido de maneira apropriada. Desta forma, uma transição entre estados ocultos representa uma mudança na distribuição dos símbolos observáveis, ou seja, cada instante de tempo em que uma transição de estado ocorre representa um ponto de mudança.

Diferentes métodos podem ser utilizados para a identificação dos estados ocultos em cada instante de tempo [61]. Escolhemos o método de Viterbi para a identifi-

cação da sequência de estados ocultos $\mathbf{S} = (s_1, s_2, ..., s_T)$ mais provável dada a série temporal de observações $\mathbf{O} = (O_1, O_2, ..., O_T)$ (Seção 3.2). A partir desta sequência, é possível detectar como pontos de mudança todos os instantes de tempo t tais que $s_t \neq s_{t-1}$.

Também propomos neste trabalho um método de detecção de pontos de mudança que funciona de maneira *online*. Na detecção *online*, a sequência de estados ocultos associados a cada amostra da série temporal é obtida através da aplicação *online* do algoritmo de Viterbi a uma janela deslizante centrada de amostras $W = \{O_{t-\alpha}, O_{t-\alpha+1}, ..., O_t, ..., O_{t+\alpha-1}, O_{t+\alpha}\}$ de tamanho $2\alpha+1$. Consequentemente, a identificação do estado oculto s_t no método *online* é realizada com um atraso de α amostras. Uma prova de conceito do método online de detecção e localização de degradação foi implementada em conjunto com uma *startup* incubada na universidade, conforme detalhado na Seção 7.1.2.

Em [74] argumenta-se que a utilização de HMMs pode gerar um número elevado de transições entre estados ocultos que não representam pontos de mudança reais. Os autores propõem a adição de um fator de regularização durante o processo de otimização dos parâmetros do modelo para aumentar a probabilidade de auto-transições, o que torna o processo de treinamento mais complexo. Propomos neste trabalho a utilização de um método mais simples, utilizando um filtro recursivo para lidar com o problema. O filtro é aplicado na sequência de estados ocultos $\mathbf{S}$, obtida através do algoritmo de Viterbi, gerando uma sequência filtrada $\mathbf{S}' = (s'_1, s'_2, ..., s'_T)$.

Formalmente, definimos o filtro da maneira que se segue. Seja $\mathbf{I}_{s_t}(s_i)$ a função indicadora definida como: $\mathbf{I}_{s_t}(s_i) = 1$ se $s_t = s_i$ e zero caso contrário. Nosso filtro tem como parâmetro de entrada o tamanho da janela de amostras δ. Primeiramente, inicializamos a sequência filtrada para o instante de tempo $t = 1$ com $s'(1) = s(1)$. Em seguida, obtemos o valor de s'_t para $t > 1$ através do seguinte procedimento:

$$s'_t = \begin{cases} s_t & \text{se } \frac{1}{\delta} \sum_{i=t-\delta+1}^{t} \mathbf{I}_{s_t}(s_i) = 1, t \geq \delta \\ s'_{t-1} & \text{caso contrário} \end{cases}$$

A partir da sequência filtrada $\mathbf{S}'$, detectamos um ponto de mudança em todo instante de tempo t em que $s'_{t-1} \neq s'_t$.

Uma das principais vantagens da utilização de HMMs é a possibilidade de interpretação do modelo treinado. Nossa abordagem se baseia na interpretação do estado oculto associado a cada amostra, o que possibilita a atribuição de um significado para diferentes segmentos da série temporal. Utilizamos a interpretação do modelo para detectar quais residências experimentam degradação na qualidade do serviço oferecido pelo ISP.

O modelo obtido após o treinamento terá associada a cada estado oculto uma

distribuição estatística para a perda de pacotes. A partir da interpretação destas distribuições definimos um conjunto de estados ocultos $\mathcal{R}$ que representam uma degradação relevante de desempenho, ou seja, estados associados à alta probabilidade de perda de pacotes ou de indisponibilidade. Em seguida, definimos um rótulo para cada série temporal diária de uma residência (par RD) de acordo com o tempo de permanência nos estados pertencentes a $\mathcal{R}$. Formalmente, definimos o rótulo $L(\mathbf{S}')$ como uma função indicadora com valor igual a 1 se o tempo de permanência nos estados pertencentes a $\mathcal{R}$ é maior do que um limiar γ e 0 caso contrário, onde γ deve ser escolhido de acordo com a qualidade de serviço almejada pelo ISP.

5.2.4 Correlação Espaço-Temporal

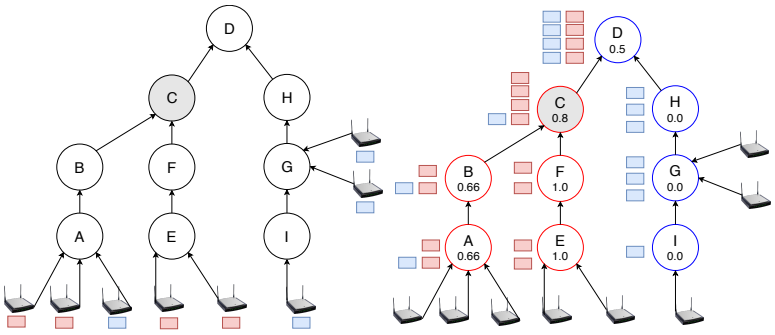
A aplicação do método de detecção baseado em pontos de mudança associa a cada par RD um rótulo $L(\mathbf{S}')$ identificando a existência (ou ausência) de degradação no desempenho da rede. O próximo passo da abordagem proposta é a correlação espaço-temporal dos resultados individuais de cada residência com o objetivo de localizar na topologia equipamentos de rede associados a degradação de desempenho.

Nosso método é baseado na suposição de que as rotas entre os pontos de medição e o servidor de medição são estáticas durante a janela de tempo considerada. Consequentemente, a topologia da rede pode ser representada através de uma árvore em que cada nó representa um ou mais equipamentos de rede do ISP. Também assumimos que a degradação de desempenho em um equipamento v afetará toda residência cuja rota até o servidor de medição passe por v. Desta forma, uma degradação de performance em v afetará o desempenho de toda a subárvore com raiz em v.

O método de correlação espaço-temporal é baseado em um esquema de votação, onde os rótulos $L(\mathbf{S}')$ obtidos por cada par RD são interpretados como votos para a qualidade do serviço de uma determinada região. Para cada nó v da topologia são considerados os votos de todos os medidores cujas rotas até o servidor de medição passem por v, ou seja, todos os medidores conectados a nós da topologia que fazem parte da subárvore de raiz v.

A técnica proposta estende o algoritmo de detecção de causa raiz de [70] ao utilizar os rótulos $L(\mathbf{S}')$ como votos e ao considerar medições realizadas em qualquer parte da topologia. Em contraste, o algoritmo de [70] se limita a medições realizadas nas folhas da árvore. A abordagem é baseada em um processo de votação majoritária no qual cada nó v recebe um rótulo $\Gamma(v)$ que indica se há degradação na qualidade do serviço. Se mais de 50% dos medidores conectados à subárvore de raiz v detectam degradação de desempenho, $\Gamma(v)$ assume valor igual a 1. Caso contrário, $\Gamma(v)$ assume valor igual a 0.

A Figura 5.1 exemplifica o método de correlação espaço-temporal proposto. O



(a) Estado inicial: séries temporais de medição e seus rótulos

(b) Processo de votação associa rótulo a cada nó

Figura 5.1: Aplicação do método de votação majoritária. Rótulos vermelhos indicam degradação de desempenho detectada.

processo de votação majoritária aplicado a cada nó v leva em consideração os rótulos $L(\mathbf{S}')$ obtidos pelas residências conectadas a equipamentos na subárvore de raiz v para identificar o rótulo $\Gamma(v)$. A partir dos resultados da correlação espaço-temporal é possível localizar grupos de nós que experimentam degradação de desempenho, como exemplifica a Figura 5.1b, possibilitando a identificação de equipamentos associados a degradação da qualidade do serviço do ISP.

5.3 Resumo

Apresentamos neste capítulo os métodos de detecção e localização de degradação não supervisionados propostos neste trabalho. Descrevemos na Seção 5.1 um novo método para detecção de degradação a partir da decomposição de um tensor formado por séries temporais multivariadas de latência e perda. Em seguida, apresentamos na Seção 5.2 um novo método de detecção e localização de degradação baseado na detecção de pontos de mudança estatística em séries temporais. Mostramos na Seção 7.1 que os métodos propostos são capazes de detectar e localizar degradações de desempenho em dados reais sem a utilização de rótulos pré-estabelecidos.

Capítulo 6

Método Baseado em Rótulos com Ruído

A obtenção de rótulos para detectar degradações de desempenho de rede representa um grande desafio para a construção de técnicas supervisionadas [1]. Motivados por este desafio, apresentamos no Capítulo 5 dois novos métodos não supervisionados para detectar degradações de desempenho na ausência de rótulos. Contudo, uma alternativa possível para identificação de degradações de desempenho são os chamados ao *call center* realizados por usuários da rede. Os registros de chamados técnicos ao *call center* podem ser prontamente obtidos pelo ISP e frequentemente estão associados a eventos que degradam a qualidade da experiência dos usuários.

No entanto, chamados realizados ao *call center* apresentam ruído inerente às percepções do usuário em relação à qualidade de sua rede de acesso. Usuários podem não abrir um chamado quando um problema acontece e vice-versa [49]. A ausência de um chamado não é um indicador confiável de boa qualidade de rede [75], uma vez que um usuário pode não perceber uma degradação na qualidade do serviço por não estar utilizando a rede no momento em que ela ocorre. Ao mesmo tempo, a degradação de desempenho de aplicações que utilizam a internet pode ser causada por eventos não relacionados a rede de acesso do ISP, como o mau funcionamento de um dispositivo do usuário ou problemas associados à rede sem fio doméstica. Os registros de reclamações ao *call center* por si só não são adequados para a detecção de degradação no desempenho da rede quando utilizados como rótulos para métodos supervisionados.

Descrevemos neste capítulo um novo método proposto para detectar e localizar eventos que degradam a qualidade da rede combinando os resultados de medições de QoS com um dataset de rótulos de QoE com ruído. Nosso método modela a condição da rede utilizando uma variável latente que não é diretamente observável e usa um modelo estatístico baseado em rótulos com ruído para inferir se uma residência está experimentando degradação na qualidade da rede. Nosso modelo é definido através

de um *Probabilistic Graphical Model* (PGM) que captura a dinâmica temporal das medições de QoS utilizando cadeias de Markov ocultas. Demonstramos como os parâmetros deste modelo podem ser estimados a partir dos dados utilizando o algoritmo de *Expectation-Maximization* (EM). Em seguida, propomos um novo método de correlação espaço-temporal para localizar degradação na rede utilizando inferência bayesiana, permitindo a estimativa de distribuições *a posteriori* que quantificam a probabilidade e a incerteza associadas a cada nó candidato problemático. Por fim, utilizamos as distribuições *a posteriori* para ranquear os problemas detectados pela correlação espaço-temporal bayesiana de maneira a facilitar o processo de *troubleshooting* do ISP.

O framework para detecção de degradação no desempenho da rede utilizando rótulos com ruído foi publicado em [14]. O detalhamento do custo computacional associado aos algoritmos utilizados por Cadeias de Markov Ocultas pode ser encontrado em [61]. Descrevemos nas seções a seguir o funcionamento das diferentes etapas do método proposto.

6.1 Pré-processamento

Exemplificamos a aplicação do método proposto utilizando séries temporais univariadas de perda como a métrica de QoS utilizada pelo modelo. Focamos na métrica de perda por conta do seu maior impacto na QoE, como mostrado nos resultados obtidos pelo método multivariado (Seção 7.1.1). Desta forma, nosso dataset é composto por pares RD extraídos a partir dos resultados de medições de perda de pacotes realizadas em roteadores residenciais a cada minuto.

A filtragem de amostras é realizada em dois níveis distintos. No nível de amostra, filtramos os resultados de medições realizadas em minutos com tráfego residencial maior do que $c_{\max}$ para filtrar o ruído associado ao tráfego residencial concorrente [25], onde $c_{\max}$ é escolhido através de um estudo empírico realizado com o conjunto de dados. Amostras filtradas por conta do tráfego concorrente são marginalizadas durante o processo de treinamento e não contribuem como evidências para o modelo. No nível de pares RD, obtemos nosso conjunto de treinamento por meio da filtragem de séries temporais diárias com pelo menos ψ amostras. Ressaltamos que a filtragem de séries temporais é aplicada apenas ao conjunto de treinamento. Durante a avaliação do método, apenas a filtragem de amostras é aplicada com o objetivo de preservar as características tipicamente encontradas em implementações no mundo real.

Além disso, intervalos de tempo sem resultados de medição são levados em consideração pelo modelo com o objetivo de identificar períodos de indisponibilidade da rede. Desta forma, codificamos estas amostras faltantes com um símbolo observável

distinto dos demais, ou seja, a ausência de uma medição é modelada como uma observação potencialmente informativa em vez de ser tratada como uma amostra faltante do ponto de vista estatístico.

A métrica de QoE com ruído utilizada como exemplo de aplicação do método é extraída através de um banco de dados com registros de chamados técnicos realizados para o *call center* do ISP parceiro (Seção 4.2). Com este dataset é possível identificar quais séries temporais de medições de QoS estão associadas a reclamações dos usuários. Esta informação é utilizada em dois momentos distintos: durante o treinamento do modelo, conforme detalhado na seção a seguir, e durante a avaliação do método bayesiano de correlação espaço-temporal (Seção 7.2.2).

6.2 Treinamento do modelo

6.2.1 Descrição do modelo

O modelo de detecção de degradação no desempenho da rede proposto é baseado no conceito de rótulos com ruído [50]. Neste cenário, desejamos classificar um conjunto de dados utilizando um rótulo Y. No entanto, o rótulo de interesse é oculto, isto é, Y representa uma variável latente, e apenas uma versão do rótulo com ruído, denotada por $\tilde{Y}$, pode ser observada. Para o cenário em que Y descreve uma variável latente binária, podemos descrever o relacionamento entre Y e o rótulo com ruído observado $\tilde{Y}$ através da seguinte probabilidade condicional:

$$\varepsilon_{m|l} \triangleq P\Big[\tilde{Y} = m \mid Y = l\Big],\ m, l \in \{0, 1\},\ \sum_{m=0}^{1} \varepsilon_{m|l} = 1. \tag{6.1}$$

Desta forma, a probabilidade marginal do rótulo com ruído divergir do rótulo latente é dada por:

$$P(\widetilde{Y} \neq Y) = \sum_{l=0}^{1} P(Y = l) \sum_{m=0}^{1} \mathbb{1}\{m \neq l\}\varepsilon_{m|l}. \tag{6.2}$$

O ruído é assimétrico, ou seja, a probabilidade de observação de um rótulo incorreto (com ruído) varia de acordo com o valor real da variável latente. Uma análise teórica do cenário de classificação com ruído binário descrito acima pode ser encontrada em [9].

Utilizamos o conceito de rótulos com ruído para propor um novo modelo estatístico de detecção de degradação de desempenho em redes em que chamados ao *call center* representam uma indicação com ruído de problema na rede. Consideramos um modelo probabilístico em que o valor de uma variável latente de interesse Y afeta a distribuição de uma série temporal de observações $O = (o_1, o_2, ..., o_T)$ de tamanho

T. Além da série temporal O, também é possível observar a versão com ruído da variável latente de interesse $\tilde{Y}$. Nosso método utiliza a distribuição conjunta das variáveis observáveis O e $\tilde{Y}$ para realizar o fit de um modelo capaz de inferir o valor da variável latente Y.

Como O representa uma série temporal, é importante capturar a dependência temporal entre as amostras. Modelamos O utilizando uma Cadeia de Markov Oculta (HMM) com o objetivo de capturar a correlação temporal entre as amostras. Desta forma, associa-se para cada observação o_t uma variável aleatória latente q_t que identifica o estado oculto (latente) no tempo t.

O nosso modelo é ilustrado através de um *Probabilistic Graphical Model* (PGM) [76] mostrado na Figura 6.1. Variáveis aleatórias observáveis são representadas por círculos preenchidos (em azul), enquanto variáveis latentes são representadas por círculos vazios. Uma seta (aresta) indica uma relação de dependência entre 2 variáveis aleatórias. Considerando que nosso dataset possui R amostras, cada variável aleatória é replicada R vezes, o que é indicado através da notação de placas (*plate notation*) e da adição de um subíndice (ou superíndice) r.

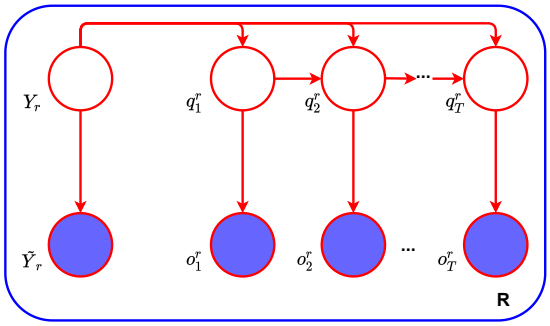


Figura 6.1: *Probabilistic Graphical Model* representando os relacionamentos entre as variáveis aleatórias propostas

A Figura 6.1 detalha as relações de dependência entre as variáveis aleatórias presentes no nosso modelo. É possível observar que a variável latente Y_r afeta tanto a distribuição da série temporal observável $\mathbf{o}^r = (o_1^r, o_2^r, ..., o_T^r)$ quanto a distribuição do rótulo com ruído observável $\tilde{Y}_r$. Consequentemente, a distribuição condicional das variáveis observáveis $\mathbf{o}^r$ e $\tilde{Y}_r$ muda de acordo com o valor da variável latente Y_r.

O PGM também é utilizado para representar a HMM que modela a série temporal observável. Desta forma, para cada observação o_t^r existe uma variável aleatória q_t^r indicando o estado oculto no tempo t. A correlação entre a série temporal observável $\mathbf{o}^r$ e o rótulo com ruído observável $\tilde{Y}_r$, capturada pelo modelo, permite a inferência do valor da variável latente de interesse Y_r a partir de um conjunto de amostras.

É importante notar que os rótulos com ruído $\tilde{Y}_r$ possuem uma granularidade diferente da série temporal de observações $\mathbf{o}^r$. No nosso dataset, uma medição observável o_t^r é coletada a cada minuto, enquanto amostras de $\tilde{Y}_r$ são coletadas

diariamente.

Descrevemos a seguir as distribuições associadas a cada variável aleatória do modelo proposto quando aplicamos o método ao nosso dataset. Nossas séries temporais de QoS são obtidas a partir das medições de perda de pacotes coletadas a cada minuto pelos roteadores residenciais. Motivados pelos padrões circadianos observados nos dados, dividimos os resultados das medições de todas as residências em R séries temporais independentes. Cada uma dessas séries temporais é denominada um par RD, indexado por r.

As séries temporais diárias possuem amostras coletadas a cada minuto. Desta forma, obtemos $T = 1440$ observações. A variável aleatória o_t^r modela a quantidade de pacotes perdidos utilizando uma distribuição categórica, onde cada amostra de o_t^r indica a quantidade de pacotes perdidos no tempo t para o par RD r. Como perdas maiores ou iguais a 20 são muito raras em nossos dados, agrupamos todas as medições de perda de 20 ou mais pacotes em uma única categoria, representada pelo símbolo 20, com o objetivo de reduzir a complexidade do modelo. Adicionalmente, instantes de tempo em que nenhum resultado de medição está disponível são codificados utilizando um símbolo especial igual a 21. Desta forma, o alfabeto de observações utilizado pelas HMMs é dado por $\mathcal{O} = \{0, \ldots, 21\}$. Para modelar a correlação temporal entre as amostras, utilizamos Cadeias de Markov Ocultas com um número discreto de estados ocultos S.

Os rótulos com ruído do nosso modelo são extraídos a partir dos chamados técnicos realizados para o *call center* do ISP. Definimos $\tilde{Y_r}$ como uma variável aleatória indicadora seguindo uma distribuição de Bernoulli, onde $\tilde{Y_r} = 1$ quando um chamado técnico realizado ao *call center* é associado ao par RD r e $\tilde{Y_r} = 0$ caso contrário. Motivados pela definição da variável $\tilde{Y_r}$, consideramos que a variável latente Y_r assume o valor $Y_r = 1$ quando a rede apresenta uma degradação na qualidade de experiência suficiente para aumentar a probabilidade de um chamado ao *call center* (informalmente, chamamos de "qualidade ruim"). Caso contrário, $Y_r = 0$ (informalmente, "qualidade boa").

Note que o modelo gráfico não define as distribuições das variáveis aleatórias, descrevendo apenas os relacionamentos entre elas. Desta forma, o PGM é geral e pode ser aplicado a qualquer problema em que uma série temporal de observações $\mathbf{o}^r$ e um rótulo com ruído $\tilde{Y_r}$ são relacionados através de uma variável latente Y_r. Por exemplo, quando consideramos modelos capazes de extrair o relacionamento entre métricas de QoS e QoE, $\mathbf{o}^r$ pode representar uma série temporal de qualquer métrica de QoS, como latência, perda e vazão, enquanto $\tilde{Y_r}$ pode representar qualquer métrica de QoE categórica ou binária, como um *score* de qualidade obtido a partir de avaliações subjetivas.

A notação utilizada é sumarizada nas tabelas 6.1 e 6.2.

r	Índice de um par Residência-Dia (RD).
$\mathcal{R}$	Conjunto de todos os pares RD em um determinado dataset.
$R = \lvert\mathcal{R}\rvert$	Número de pares RD no dataset.
t	Índice do minuto em um dia, $t = 1, \ldots, T$.
T	Tamanho da série temporal diária; $T = 1440$.
S	Número de estados ocultos da cadeia.
j, k	Índices dos estados ocultos, $j, k \in \{0, \ldots, S-1\}$.
$\lvert\mathcal{O}\rvert$	Tamanho do alfabeto de observações; $\lvert\mathcal{O}\rvert = 22$.
$\mathcal{O} = \{0, \ldots, 21\}$	Alfabeto de observações. Símbolo 20 agrega "$\geq$ 20 pacotes perdidos"; símbolo 21 representa uma medição ausente.
o	Símbolo de observação genérico, $o \in \mathcal{O}$.
l, m	Valores de Y_r e $\tilde{Y}_r$ respectivamente, $l, m \in \{0, 1\}$, e índices na Equação (6.1).
l, l'	Índices na Equação (6.5).
d	Índice do dia na avaliação (Seção (7.2.2)).
$\mathcal{D}$	Conjunto de dias sobre os quais avaliações são agregadas.
$c_{\max}$	Limiar de tráfego concorrente para filtrar amostras.
$\mathbf{o}^r$	Sequência de observações de perda $(o_1^r, \ldots, o_T^r)$ para o par RD r.
$\mathbf{q}^r$	Sequência de estados ocultos $(q_1^r, \ldots, q_T^r)$ para o par RD r.
o_t^r	Símbolo observável representando o número de pacotes perdidos em um trem de 100 no minuto t.
q_t^r	Estado oculto no minuto t.
Y_r	Qualidade latente da rede; $Y_r = 1$ denota degradação.
$\tilde{Y}_r$	Rótulo com ruído; $\tilde{Y}_r = 1$ se chamado técnico é associado ao par RD r.
$\hat{Y}_r$	Estimativa MAP de Y_r.
$\mathbf{Y}, \tilde{\mathbf{Y}}, \mathbf{O}, \mathbf{q}$	Coleções sobre todo r (Apêndice A.1).

Tabela 6.1: Notação: Índices, conjuntos, dimensões e variáveis aleatórias

ρ_l	Probabilidade a priori da classe $P[Y_r = l]$.
$\varepsilon_{m\lvert l}$	Modelo para chamados observados $P[\tilde{Y}_r = m \mid Y_r = l]$.
$\pi_k^{(l)}$	Distribuição inicial dos estados ocultos $P[q_1^r = k \lvert Y_r = l]$
$a_{jk}^{(l)}$	Probabilidade de transição $P[q_{t+1}^r = k \mid q_t^r = j,\, Y_r = l]$.
$b_j^{(l)}(o)$	Probabilidade de emissão $P[o_t^r = o \mid q_t^r = j,\, Y_r = l]$.
$\theta, \theta^{\text{old}}$	Vetor de parâmetros $\theta = \{\rho_l, \varepsilon_{m\lvert l}, \pi_k^{(l)}, a_{jk}^{(l)}, b_j^{(l)}(o)\}$.

Tabela 6.2: Notação: Parâmetros do modelo.

6.2.2 Estimativa dos parâmetros do modelo

O PGM proposto (Figura 6.1) contém um conjunto de variáveis aleatórias latentes e um conjunto de variáveis aleatórias observáveis. Consequentemente, é possível maximizar a verossimilhança do modelo utilizando o algoritmo de Expectation-Maximization (EM) [77], isto é, podemos utilizar o algoritmo de EM para encontrar

os parâmetros do modelo que maximizam a probabilidade de observar nosso conjunto de amostras. A seguir apresentamos as equações necessárias para a aplicação do algoritmo de EM ao modelo proposto. A derivação destas equações pode ser encontrada no Apêndice A.1.

Considerando as distribuições das variáveis aleatórias descritas na Seção 6.2.1, o modelo possui o conjunto de parâmetros a serem estimados descrito na Tabela 6.2. Para estimar estes parâmetros, definimos um conjunto de variáveis auxiliares relacionadas à distribuição das variáveis latentes descritas na Tabela 6.3.

$\alpha_{r,t}^{(l)}(j)$	$P[o_1^r, o_2^r, \ldots, o_t^r,\ q_t^r = j \mid Y_r = l]$
$\beta_{r,t}^{(l)}(j)$	$P[o_{t+1}^r, o_{t+2}^r, \ldots, o_T^r \mid q_t^r = j,\ Y_r = l]$
$\gamma_{r,t}^{(l)}(j)$	$P[q_t^r = j \mid \mathbf{o}^r,\ Y_r = l]$
$\xi_{r,t}^{(l)}(j,k)$	$P[q_t^r = j,\ q_{t+1}^r = k \mid \mathbf{o}^r,\ Y_r = l]$
$\tau_r(l)$	$P[Y_r = l \mid \mathbf{o}^r,\ \tilde{Y}_r]$

Tabela 6.3: Notação: Variáveis auxiliares

Utilizamos o modelo de EM para estimar os parâmetros do modelo (Tabela 6.2) a partir das variáveis auxiliares (Tabela 6.3). No passo de *Expectation*, o valor das variáveis $\alpha_{r,t}^{(l)}(j), \beta_{r,t}^{(l)}(j), \gamma_{r,t}^{(l)}(j), \xi_{r,t}^{(l)}(j,k)$ é calculado utilizando as variáveis do algoritmo de *Forward-Backward* (descritas na Seção 3.2), enquanto $\tau_r(l)$ (definido acima) é dado por:

$$\tau_r(l) = \frac{\left(\sum_{j=0}^{S-1} \alpha_{r,T}^{(l)}(j)\right) \varepsilon_{\tilde{Y}_r|l}\ \rho_l}{\sum_{u=0}^{1} \left(\sum_{j=0}^{S-1} \alpha_{r,T}^{(u)}(j)\right) \varepsilon_{\tilde{Y}_r|u}\ \rho_u}. \tag{6.3}$$

Onde $\tau_r(l)$ pode ser interpretado como a responsabilidade [76] associada à classe l para o par RD r.

Os parâmetros de interesse são atualizados a partir das variáveis auxiliares no passo de *Maximization*. Utilizando o algoritmo de EM, obtemos:

$$\begin{aligned}
&\rho_l = \frac{\bar{\tau}_l}{R} \qquad \varepsilon_{m|l} = \frac{\sum_{r=1}^{R} \tau_r(l)\, \mathbb{1}\{\tilde{Y}_r = m\}}{\bar{\tau}_l} \\
&\pi_k^{(l)} = \frac{\sum_{r=1}^{R} \tau_r(l)\, \gamma_{r,1}^{(l)}(k)}{\bar{\tau}_l} \qquad a_{jk}^{(l)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T-1} \tau_r(l)\, \xi_{r,t}^{(l)}(j,k)}{\sum_{r=1}^{R} \sum_{t=1}^{T-1} \tau_r(l)\, \gamma_{r,t}^{(l)}(j)} \\
&b_j^{(l)}(o) = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T} \tau_r(l)\, \gamma_{r,t}^{(l)}(j)\, \mathbb{1}\{o_t^r = o\}}{\sum_{r=1}^{R} \sum_{t=1}^{T} \tau_r(l)\, \gamma_{r,t}^{(l)}(j)}
\end{aligned} \tag{6.4}$$

Onde $\bar{\tau}_l = \sum_{r=1}^{R} \tau_r(l)$ representa o número efetivo de pares RD associados à

classe l.

6.3 Detecção de degradação

O nosso modelo tem como objetivo principal inferir a qualidade de rede latente Y_r associada a um par RD r a partir de sua série temporal de medições de QoS $\mathbf{o}^r$, permitindo a detecção automática de degradações de desempenho de rede. Após estimar os parâmetros do modelo utilizando o algoritmo de EM (Seção 6.2.2), o processo de inferência utiliza a Regra de Bayes:

$$P\left[Y_r = l \mid \mathbf{o}^r\right] = \frac{\mathcal{L}_l(\mathbf{o}^r)\,\rho_l}{\sum_{l'=0}^{1} \mathcal{L}_{l'}(\mathbf{o}^r)\,\rho_{l'}} \tag{6.5}$$

Onde $\mathcal{L}_l(\mathbf{o}^r) = P[\mathbf{o}^r \mid Y_r = l] = \sum_{j=0}^{S-1} \alpha_{r,T}^{(l)}(j)$ é obtido a partir do algoritmo de Forward (Seção 3.2). Utilizamos a Equação 6.5 para calcular a probabilidade de um evento que degrada a qualidade da rede dada uma série temporal de medições de QoS, isto é, $P[Y_r = 1 \mid \mathbf{o}^r]$.

Nossa aplicação de exemplo utiliza a Equação 6.5 em séries temporais de QoS diárias, uma vez que os rótulos com ruído obtidos pelo ISP têm granularidade diária. Entretanto, ressaltamos que o modelo poderia ser treinado em escalas de tempo menores, desde que a escala de tempo dos rótulos obtidos fosse inferior a um dia.

Note que os rótulos com ruído $\tilde{Y}_r$ não estão presentes na Equação 6.5, ou seja, a informação de chamados ao call center é usada durante o treinamento do modelo, mas não é necessária para a detecção de degradação de desempenho da rede. Além disso, a aplicação da regra de Bayes pode ser realizada a partir de uma série temporal de observações de qualquer tamanho, isto é, o método pode ser aplicado de maneira *online* após o treinamento do modelo sem a necessidade de nenhuma alteração na Equação 6.5.

6.4 Correlação Espaço-Temporal

Descrevemos nesta seção o algoritmo de correlação espaço-temporal bayesiano proposto para localizar degradações de desempenho de rede. Nossa abordagem leva em consideração as detecções obtidas através do modelo apresentado na Seção 6.2.1 e a topologia de rede do ISP para inferir quais equipamentos de rede experimentam degradação de desempenho. O método usa inferência bayesiana para estimar a probabilidade *a posteriori* de falha associada a cada nó potencialmente problemático, possibilitando uma localização de falhas capaz de estimar o nível de certeza associado a cada detecção.

Detalhamos a seguir as suposições consideradas pelo método proposto. Assumimos que as rotas entre cada dispositivo de medição e o servidor de medição são estáticas durante o intervalo de medição, uma vez que este é o caso observado no nosso dataset (note que o método pode ser estendido para um cenário com rotas dinâmicas ao se associar uma probabilidade para cada rota, como em [22]). Consequentemente, a topologia da rede pode ser representada por uma árvore $\mathcal{G} = (V, E)$ com um conjunto de nós V e um conjunto de arestas E, onde cada nó $v \in V$ representa um ou mais equipamentos de rede. Também consideramos que uma degradação de desempenho causada pelo nó v afeta todas as residências cujas rotas na rede passam por v, isto é, todos os clientes conectados a nós na subárvore de raiz v experimentarão uma degradação no desempenho sempre que v tiver um problema.

O método de correlação espaço-temporal proposto utiliza o modelo de detecção descrito na Seção 6.2.1 para identificar quais residências experimentam degradação de desempenho na sua rede. Para cada residência com um medidor, calculamos a probabilidade da qualidade latente da rede ser "ruim" dada uma série temporal de medições de QoS ($P[Y_r = 1 \mid \mathbf{o}^r]$) utilizando a Equação 6.5. Baseado nos resultados obtidos, associamos a cada par RD $= r$ o valor mais provável para a qualidade latente da rede Y_r, denotando esta estimativa como $\hat{Y_r}$. Formalmente, definimos $\hat{Y_r}$ como:

$$\hat{Y_r} = \mathbb{1}\left\{P[Y_r = 1 \mid \mathbf{o}^r] > P[Y_r = 0 \mid \mathbf{o}^r]\right\}. \tag{6.6}$$

Em seguida, correlacionamos os resultados obtidos por todas as residências com medidores para identificar equipamentos com degradação de desempenho. Especificamente, para identificar se um nó v da topologia apresenta problemas de desempenho, correlacionamos os resultados do modelo $\hat{Y_r}$ obtidos por todas as residências cujas rotas na rede passam pelo nó v. Intuitivamente, espera-se um número alto de detecções de degradação em residências conectadas à subárvore de raiz v sempre que o nó v apresentar degradação de desempenho. Além disso, obtemos mais certeza sobre os resultados da nossa inferência à medida que o tamanho da amostra aumenta.

Formalizamos estas intuições a partir de um método de correlação espaço-temporal baseado em inferência bayesiana. A partir dos resultados deste método, é possível inferir a incerteza associada a um problema detectado em um nó v baseado no número de detecções e no tamanho da amostra. Especificamente, associa-se a cada nó v uma variável aleatória ϕ_v representando a probabilidade de que um par RD extraído de uma residência conectada à subárvore de raiz v seja classificado como degradado. Para inferir ϕ_v, levamos em consideração o número de pares RD extraídos de residências conectadas à subárvore de raiz v, denotado por n_v, e o

número de detecções entre esses pares RD, denotado por F_v. Por exemplo, para a topologia representada na Figura 6.2, temos $n_A = 7$ e $F_A = 3$ para o nó $v = A$, enquanto $n_B = 4$ e $F_B = 0$ para o nó $v = B$.

Formalmente, definimos $\mathcal{R}_v$ como o conjunto de pares RD extraídos a partir de residências conectadas à subárvore de raiz v e definimos F_v e n_v como:

$$F_v = \sum_{r \in \mathcal{R}_v} \widehat{Y}_r, \qquad n_v = |\mathcal{R}_v|.$$

Utilizamos a distribuição *a posteriori* de ϕ_v para inferir se o nó v está experimentando degradação de desempenho. A intuição por trás do método é que uma fração alta de dispositivos de medição conectados à subárvore de raiz v detectará degradação de desempenho sempre que o nó v tiver um problema. Consequentemente, é esperado que ϕ_v assuma valores mais altos com alta probabilidade sempre que o nó v tiver um problema de desempenho. Além disso, também é possível medir o grau de confiança nos nossos resultados a partir da distribuição *a posteriori* de ϕ_v, obtida utilizando inferência bayesiana.

Baseado nestas observações, consideramos que um nó v apresenta um problema de desempenho sempre que ϕ_v apresentar um valor maior que um limiar δ com um grau de confiança maior que η. Formalizamos esta ideia através da definição de uma variável aleatória indicadora Z_v que identifica a existência de degradação de desempenho no nó v, definida por:

$$Z_v = \mathbb{1}\left\{P[\phi_v > \delta \mid F_v, n_v] > \eta\right\} \tag{6.7}$$

Para obter o valor de Z_v é necessário inferir a distribuição *a posteriori* de ϕ_v, isto é, precisamos calcular o valor de $P[\phi_v | F_v, n_v]$. Consideramos que a distribuição *a priori* de ϕ_v segue uma distribuição beta com parâmetros α_0 e β_0, enquanto a distribuição do número de detecções F_v é modelada utilizando uma distribuição binomial com parâmetros ϕ_v e n_v. Consequentemente, a distribuição *a posteriori* para a variável aleatória ϕ_v é dada por uma distribuição beta com parâmetros $\alpha_0 + F_v$ e $\beta_0 + n_v - F_v$, uma vez que a distribuição beta é a prior conjugada da distribuição binomial [78]. Consequentemente, obtemos uma expressão fechada para a distribuição *a posteriori* de ϕ_v, o que possibilita um cálculo computacionalmente eficiente para o valor de Z_v para qualquer nó v. A Tabela 6.4 contém um resumo da notação adicional usada na definição do método de correlação espaço-temporal.

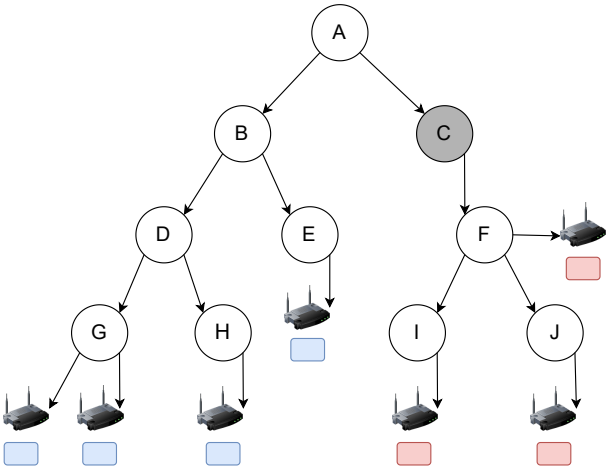
O cenário em que um problema no nó v afeta todos os clientes na subárvore de raiz v torna possível a aplicação de um algoritmo de busca para identificar de maneira eficiente quais equipamentos de rede experimentam degradação de desempenho. Nosso algoritmo percorre a topologia em árvore $\mathcal{G}$ utilizando uma busca

$\mathcal{G} = (V, E)$	Árvore representando rede de acesso.
v	Nó da topologia; $v \in V$.
$\mathcal{R}_v$	Pares RD cujo caminho de medição passa pela subárvore com raiz v; $\mathcal{R}_v \subseteq \mathcal{R}$.
ϕ_v	Parâmetro da likelihood binomial para F_v: a probabilidade de que um par RD monitorado na subárvore com raiz em v seja classificado como degradado.
n_v	$n_v = \lvert\mathcal{R}_v\rvert$: pares RD monitorados na subárvore com v.
F_v	Número de detecções na subárvore com v. $F_v = \sum_{r \in \mathcal{R}_v} \hat{Y}_r$.
Z_v	Indicador que v é marcado como problemático.
δ	Limiar em ϕ_v. (Usamos $\delta = 0.75$.)
η	Confiança necessária para marcar um nó extraída da distribuição a posteriori. (Usamos $\eta = 0.95$.)
α_0, β_0	Hiperparâmetros da distribuição a priori Beta. (Usamos $\alpha_0 = \beta_0 = 1$.)
$\mathcal{A}$	Conjunto de nós potencialmente afetados.
$\bar{\mathcal{A}}$	Conjunto complementar: nós que não representam raízes marcadas nem descendentes de raízes marcadas.
$\mathcal{A}^\star$	Conjunto mínimo de raízes de subárvores afetadas.
$\mathcal{A}_d^{\text{hi}}$, $\mathcal{A}_d^{\text{lo}}$	Subárvores com raízes ranqueadas no top-5 e subárvores com raízes fora do top-5 no dia d.
κ_v	$(1-\zeta)$ quantil da distribuição a posteriori de ϕ_v; estatística de ranqueamento.
ζ	Nível de confiança para ranqueamento; $\zeta = 0.95$.

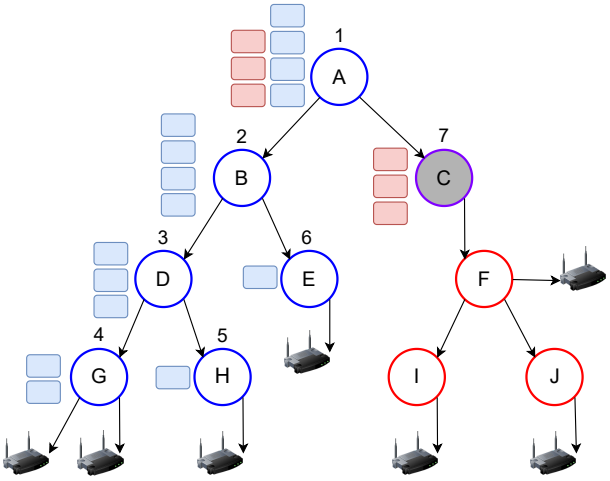
Tabela 6.4: Notação: Correlação Espaço-Temporal.

em profundidade para identificar quais nós experimentam problemas de desempenho. Para cada nó v visitado durante a busca, aplica-se o algoritmo de correlação detalhado acima para detectar potenciais problemas na rede. Se o nosso algoritmo de correlação identifica um problema no nó v (isto é, $Z_v = 1$), então todos os nós abaixo de v são afetados pelo problema. Desta forma, aplica-se uma poda na busca sempre que um nó v é considerado problemático pelo algoritmo de correlação.

Definimos todos os nós pertencentes à subárvore de um nó marcado como problemático pelo nosso método como *potencialmente afetados* por um problema e denotamos este conjunto de nós como $\mathcal{A} \subseteq V$. O conjunto complementar de $\mathcal{A}$ é definido como $\bar{\mathcal{A}} = V \setminus \mathcal{A}$, isto é, o conjunto de nós que não representam raízes marcadas nem descendentes de raízes marcadas. Definimos o subconjunto de nós identificados como raízes de subárvores em $\mathcal{A}$ como $\mathcal{A}^\star \subseteq \mathcal{A}$. O subconjunto $\mathcal{A}^\star$ representa o conjunto de nós problemáticos que poderia explicar todos os problemas de desempenho localizados pelo nosso algoritmo de correlação. A Figura 6.2 exemplifica o método de busca executado pelo nosso algoritmo de correlação espaço-temporal e os conjuntos $\mathcal{A}$ e $\bar{\mathcal{A}}$ obtidos como resultado de sua aplicação.



(a) Resultados do método de detecção de degradação. Retângulos vermelhos representam uma detecção de degradação ($\hat{Y}_r = 1$) enquanto retângulos azuis representam ausência de detecção ($\hat{Y}_r = 0$)



(b) Correlação espaço-temporal divide os nós da árvore nos subconjuntos $\mathcal{A}^\star$ (roxo), $\mathcal{A}$ (vermelho) e $\bar{\mathcal{A}}$ (azul). Números indicam a ordem na qual os nós são visitados

Figura 6.2: Exemplo de correlação espaço-temporal ($\delta = 0.45$, $\eta = 0.95$). Nó C apresenta problemas e é responsável pela degradação do desempenho observada.

6.5 Ranqueamento de degradações

Os recursos operacionais dedicados para mitigar degradações de desempenho da rede em um ISP são frequentemente limitados. Desta forma, é essencial priorizar esforços durante o processo de *troubleshooting* da rede. Propomos um algoritmo de ranqueamento que produz uma ordenação para os problemas detectados com o algoritmo de correlação espaço-temporal bayesiano com o objetivo de melhorar o processo de correção de falhas na rede do provedor de serviço.

Nosso método de ranqueamento utiliza a distribuição *a posteriori* de ϕ_v (isto

é, $P[\phi_v|F_v, n_v]$) para obter uma ordenação do conjunto de nós $v \in \mathcal{A}^\star$, isto é, o algoritmo provê uma ordenação para o conjunto de nós que poderiam explicar todos os problemas de degradação de desempenho localizados pelo método de correlação espaço-temporal. Lembramos que a variável aleatória ϕ_v representa a prevalência de detecções de degradação de desempenho em medidores conectados a subárvore de raiz v, que é desconhecida, enquanto F_v/n_v representa a fração observada correspondente. Um método de ranqueamento baseado apenas na média da distribuição *a posteriori* de ϕ_v não seria capaz de levar em consideração a incerteza associada a estimativa desta variável, que varia de maneira considerável de acordo com o número n_v de pares RD monitorados. Desta forma, a ideia por trás do nosso método de ranqueamento é extrair da distribuição *a posteriori* de ϕ_v uma estatística que ordene os resultados obtidos por cada nó $v \in \mathcal{A}^\star$ com um certo grau de confiança.

Consideramos como estatística de ordenação o quantil $1 - \zeta$ da distribuição *a posteriori* de ϕ_v, denotada por κ_v, para ordenar os problemas detectados, definida como:

$$P[\phi_v > \kappa_v|F_v, n_v] = \zeta \tag{6.8}$$

Podemos interpretar κ_v como um limiar inferior para o valor de ϕ_v quando consideramos um grau de confiança ζ, isto é, temos um grau de confiança ζ de que o valor de ϕ_v é maior que κ_v. Desta forma, κ_v provê uma medida conservadora da prevalência da degradação.

Nosso algoritmo de ranqueamento ordena todos os nós problemáticos $v \in \mathcal{A}^\star$ de acordo com o valor de κ_v. Nossos resultados (Capítulo 7) mostram que o algoritmo de ranqueamento proposto pode auxiliar o ISP a identificar problemas crônicos na rede com um alto impacto na QoE dos usuários.

6.6 Resumo

Apresentamos neste capítulo um novo método de detecção e localização de degradação de desempenho de rede baseado em medições de QoS e rótulos de QoE com ruído. A Seção 6.2 descreve o modelo estatístico proposto e o processo de treinamento do modelo. Mostramos como o modelo pode ser utilizado para identificar degradações na Seção 6.3. Propomos um novo método de correlação espaço-temporal dos resultados do modelo de detecção baseado em inferência bayesiana na Seção 6.4. Por fim, descrevemos um método de ranqueamento de problemas detectados na Seção 6.5 com o objetivo de auxiliar o ISP durante o processo de *troubleshooting*. Nossos resultados, descritos na Seção 7.2, evidenciam a utilidade do modelo na identificação automática de degradações de desempenho que afetam a qualidade de experiência

dos usuários do ISP.

Capítulo 7

Resultados

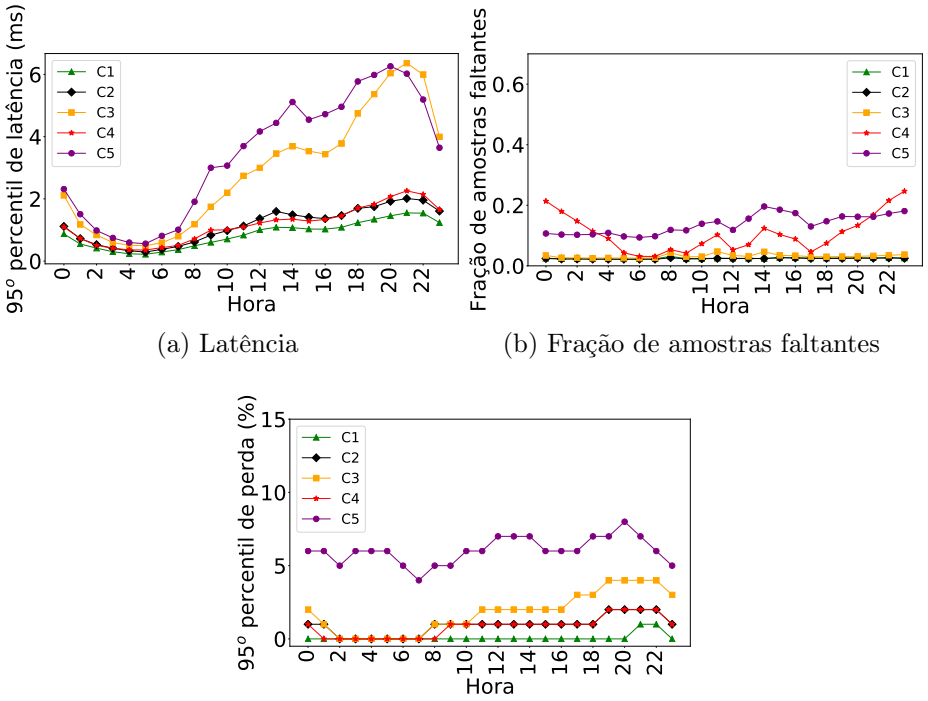
Descrevemos neste capítulo os resultados obtidos pelas técnicas propostas neste trabalho. A Seção 7.1 apresenta os resultados alcançados pelos métodos não supervisionados (Capítulo 5). Mostramos na Seção 7.1.1 que o framework baseado nos residuais da decomposição tensorial de PARAFAC é capaz de detectar diferentes eventos de rede sem a necessidade de rótulos. Posteriormente, a Seção 7.1.2 mostra que o nosso método baseado na detecção de pontos de mudança é capaz de localizar regiões da rede que experimentam degradação de desempenho e de realizar a detecção de degradação de maneira *online*. Também descrevemos a implementação de uma prova de conceito que mostra a viabilidade da aplicação do método *online* de detecção de pontos de mudança proposto em um cenário real.

Por fim, apresentamos na Seção 7.2 os resultados obtidos pelo método baseado em medições de QoS e rótulos de QoE com ruído (Capítulo 6). Nossos resultados mostram que a probabilidade de criação de um chamado ao call center é 4,5 vezes maior quando um problema é detectado em comparação com cenários sem detecção pelo nosso modelo. Além disso, mostramos que o método de ranqueamento é capaz de auxiliar no processo de *troubleshooting* e apresentamos exemplos de eventos reais de degradação detectados pelo nosso método.

7.1 Métodos não supervisionados

7.1.1 Método baseado em Decomposição Tensorial

Mostramos nesta seção que o método de detecção baseado em decomposição tensorial é capaz de identificar eventos reais de rede sem a necessidade de rótulos de degradação. Consideramos como entrada para o ajuste do modelo de decomposição tensorial os resultados das medições de latência e perda realizadas por 2964 roteadores domésticos entre os dias 19/08/2019 e 22/09/2019. Definimos o limiar de tráfego concorrente $\theta = 2.5$ Mbps a partir dos resultados de um estudo empírico



(a) Latência

(b) Fração de amostras faltantes

(c) Perda

Figura 7.1: Séries temporais sumarizadas por cluster

realizado com o nosso conjunto de dados, no qual observamos que um tráfego acima deste limiar gerava perda e latência causadas pelo tráfego residencial. Consideramos nesta análise séries temporais com ao menos $\psi = 1000$ amostras.

Após a aplicação do pré-processamento, foram obtidas 50282 séries temporais multivariadas, em que cada série temporal possui até 1440 amostras diárias. Desta forma, o treinamento do modelo é realizado utilizando um tensor de terceira ordem $\mathcal{X} \in \mathbb{R}^{50282 \times 2 \times 1440}$. Após a aplicação do método de *Split-Half Validation*, definimos o número de fatores $R = 4$. Por fim, utilizamos o método do cotovelo para escolher o número de clusters $K = 5$.

Resultados da clusterização

Para entender o comportamento dos clusters obtidos após a aplicação do modelo de Mistura de Gaussianas, realizamos a análise de sumários das séries temporais de medições atribuídas a cada cluster. Consideramos a sumarização de três métricas: perda, fração de amostras faltantes e latência. Para a análise da métrica de latência, subtraímos de cada série temporal o seu valor mais baixo com o objetivo de inferir o tempo de enfileiramento de pacotes durante períodos de congestionamento.

A Figura 7.1 mostra o resultado da sumarização de cada uma das métricas consideradas. É possível visualizar uma interpretação diferente para cada cluster obtido. O cluster C1 é relacionado a pares RD com a melhor qualidade de serviço (baixa latência, baixa perda, baixa indisponibilidade). O cluster C2 representa pares RD

que experimentam perdas moderadas, mas que possuem baixa latência e pouca indisponibilidade. Pares RD com latência alta e perdas moderadas são associados ao cluster C3. O cluster C4 representa séries temporais com períodos de alta indisponibilidade. Por fim, o cluster C5 representa pares RD com alta latência, alta perda e alta indisponibilidade. É importante frisar que a clusterização é realizada a partir de pares RD, ou seja, uma mesma residência pode ser associada a diferentes clusters em diferentes dias.

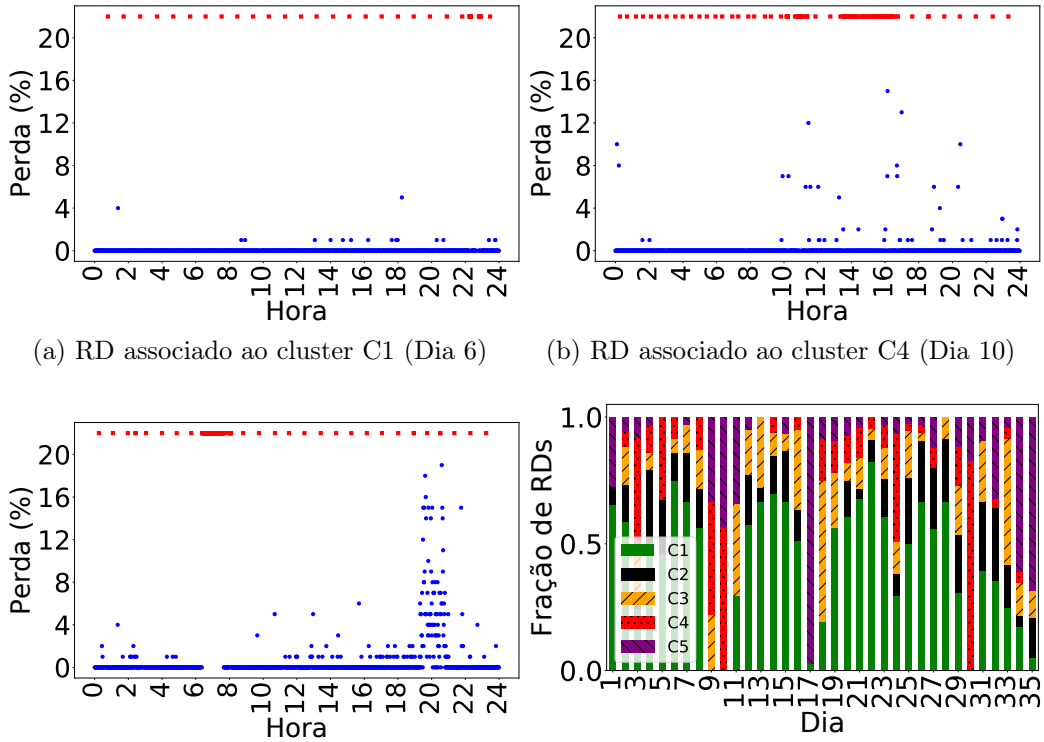
Resultados da correlação espaço-temporal

Possuímos em nosso *dataset* a localização dos roteadores residenciais que realizam medição na topologia do ISP, como detalhado na Seção 4.2. Desta forma, podemos correlacionar espacialmente os resultados das residências de acordo com a sua proximidade geográfica com o intuito de identificar eventos de rede que afetem múltiplas residências de maneira simultânea. É esperado que residências que compartilhem as mesmas rotas experimentem um desempenho de rede similar em termos de congestionamentos e falhas. Desta forma, agrupamos as residências de acordo com as regiões da rede do ISP (definidas na Seção 4.2.2).

Ilustramos os resultados da correlação espaço-temporal para uma determinada região de rede na Figura 7.2. Resultados similares podem ser encontrados para outras regiões. É possível observar que a maioria das residências foi associada ao cluster C1 (boa qualidade de serviço) na maior parte dos dias. A Figura 7.2a exemplifica a série temporal de perda de pacotes de uma residência associada ao cluster C1 no dia 6. Note que a fração de perda é próxima de zero durante a maioria dos minutos do dia.

Por outro lado, uma grande quantidade de residências é associada aos clusters C4 e C5 no dia 10. Observam-se neste dia múltiplas séries temporais de medições com amostras faltantes entre 13:00 e 17:00, como demonstrado na Figura 7.2b. Outro tipo de evento detectado pela correlação espaço-temporal ocorre no dia 17, quando múltiplas séries temporais apresentam amostras faltantes entre 6:00 e 8:00 e altas perdas entre 19:00 e 21:00. Estas séries temporais são associadas ao cluster C5 no dia 17. Este comportamento é ilustrado pelos resultados de medições de perda mostrados na Figura 7.2c.

A Figura 7.3 exemplifica o resultado da correlação espaço-temporal obtido após a aplicação do método de clusterização em duas regiões da rede. É possível observar que ambas possuem desempenho bastante distinto. A região mostrada na Figura 7.3a frequentemente possui uma alta fração de residências associadas ao cluster C1, representando a melhor qualidade de rede. Por outro lado, é raro observar uma residência associada ao cluster C1 na região mostrada na Figura 7.3b. Estes resultados exemplificam como o nosso método pode ser utilizado para identificar quais



(a) RD associado ao cluster C1 (Dia 6)

(b) RD associado ao cluster C4 (Dia 10)

(c) RD associado ao cluster C5 (Dia 17)

(d) Fração de RDs por dia.

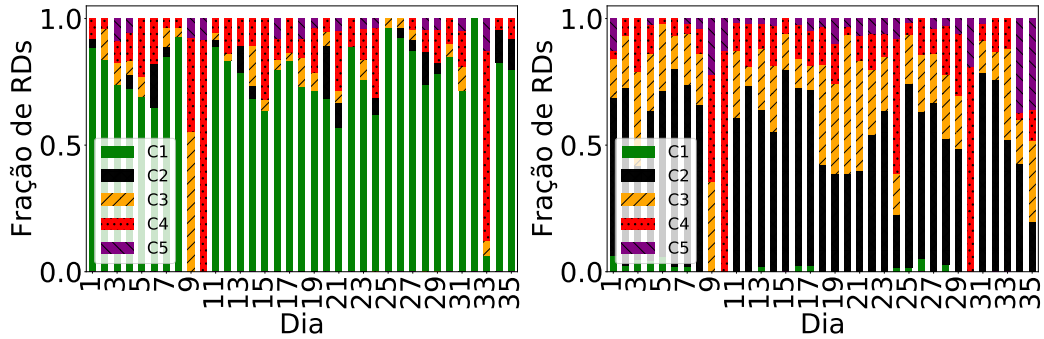
Figura 7.2: Exemplo de correlação espaço-temporal (amostras faltantes são representadas utilizando quadrados vermelhos).

regiões da rede possuem pior desempenho, auxiliando o ISP durante a detecção e localização de problemas na rede.

Eventos de rede detectados

Além de identificar regiões da rede com melhor desempenho, os resultados da correlação espacial também permitem a detecção não supervisionada de eventos reais de rede. Exemplificamos a seguir eventos de rede que foram detectados de maneira não supervisionada pelo nosso método e que tiveram a causa raiz identificada com o auxílio do ISP. Para a detecção dos eventos de rede aplicamos o método proposto aos resultados de medições obtidos entre os dias 16/01/2020 e 31/12/2020. Após a filtragem, foram obtidos 1282140 pares RD de 6307 residências distintas. Note que os resultados apresentados a seguir consideram os mesmos clusters descritos anteriormente (Figura 7.1).

Quatro eventos distintos detectados pela nossa abordagem não supervisionada são exibidos na Figura 7.4. A Figura 7.4a mostra os resultados obtidos pela clusterização de todas as residências do *dataset* durante o mês de março de 2020. É possível observar que todos os pares RD foram associados ao cluster C5 nos dias 26/03/2020 e 27/03/2020. Este cluster é associado a períodos de indisponibilidade na rede. Nestes dois dias, o ISP foi alvo de um ataque de negação de serviço distribuído (DDoS) que causou períodos de indisponibilidade na rede de seus usuários. Nosso método foi



(a) Região A (b) Região B

Figura 7.3: Desempenho da rede obtido a partir da clusterização dos resíduos

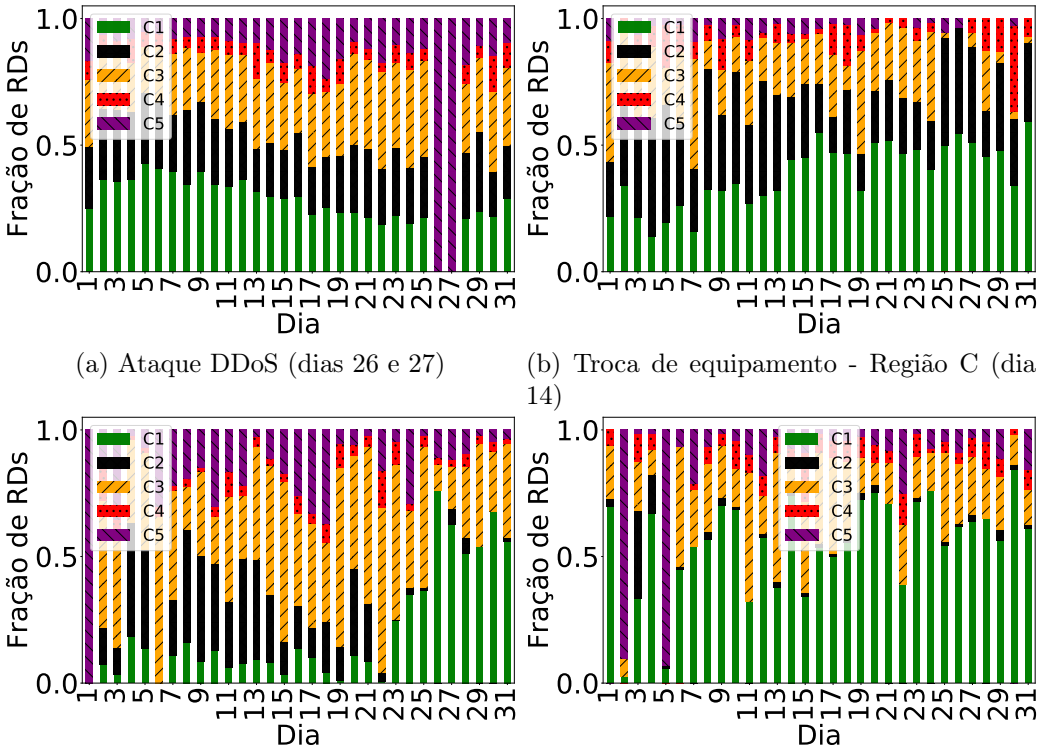
capaz de detectar automaticamente o ataque a partir da degradação do desempenho causada pelo evento.

A clusterização dos residuais da decomposição tensorial também é capaz de identificar melhorias na qualidade do serviço ocasionadas por intervenções na rede realizadas pelo ISP. A Figura 7.4b mostra que a fração de usuários associados ao cluster C1 (melhor qualidade) aumenta a partir do dia 14/05. O ISP realizou a troca de um equipamento durante o período, o que explica a melhora observada. Outro exemplo de melhoria é mostrado na Figura 7.4c, onde é possível observar um aumento na fração de usuários associados ao cluster C1 após o dia 23/05. O ISP iniciou durante este período um processo de reconfiguração da rede desta região, melhorando a qualidade do serviço oferecido aos seus clientes.

Nosso método também é capaz de identificar falhas em equipamentos. A Figura 7.4d mostra os resultados da clusterização para uma região específica da rede durante o mês de agosto de 2020. É possível observar que uma grande fração de usuários é associada ao cluster C5 (relacionado à indisponibilidade da rede) nos dias 02/08/2020 e 05/08/2020. Uma falha elétrica em um equipamento ocorreu no dia 02/08, tornando a rede indisponível por algumas horas. A troca do equipamento foi realizada no dia 05/08. Ambos os eventos foram identificados de maneira não supervisionada pelo método proposto.

Além de eventos diários, nosso método também possibilita a detecção de tendências de desempenho em períodos mais longos de tempo. Analisamos os resultados obtidos no nosso dataset entre os dias 01/03/2020 e 31/07/2020, mostrados na Figura 7.5. É possível observar uma degradação no desempenho da rede no meio de março, quando a fração de residências associadas ao cluster de melhor qualidade de serviço cai pela metade. O desempenho melhora lentamente durante os meses subsequentes, quando mais residências voltam a ser associadas ao cluster C1.

Uma explicação plausível para a tendência observada é a mudança do comportamento dos usuários durante o período de confinamento causado pela pandemia de COVID-19. O período de confinamento foi iniciado a partir do dia 16/03/2020 e as



(a) Ataque DDoS (dias 26 e 27)

(b) Troca de equipamento - Região C (dia 14)

(c) Reconfiguração da rede - Região D (dia 23)

(d) Manutenção na rede - Região E (dia 2 e dia 5)

Figura 7.4: Eventos detectados pelo nosso método

restrições foram flexibilizadas gradualmente nas semanas subsequentes.

Relação entre clusters e QoE

Para avaliar a relação entre os clusters obtidos pelo nosso método de detecção não supervisionada e a qualidade de experiência dos usuários, correlacionamos os resultados da clusterização obtidos entre os dias 16/01/2020 e 31/12/2020 com os chamados técnicos realizados ao call center. A Figura 7.6a mostra, para cada cluster, a razão entre o número de RDs associados a um chamado ao call center e o número total de RDs no cluster (multiplicada por 100). É possível observar que os clusters C4 e C5 apresentam a maior porcentagem de RDs com chamados ao call center. Ao mesmo tempo, os clusters C2 e C3 têm frações menores de RDs com chamados.

Note que clientes associados ao cluster de melhor qualidade (C1) também realizam chamados ao call center, exemplificando o ruído associado a este tipo de rótulo. Ao mesmo tempo, é possível observar que a latência apresenta menor impacto na QoE dos usuários, uma vez que o cluster associado à alta latência (C3) apresenta uma fração de chamados uma ordem de grandeza menor que a observada para os clusters C4 e C5.

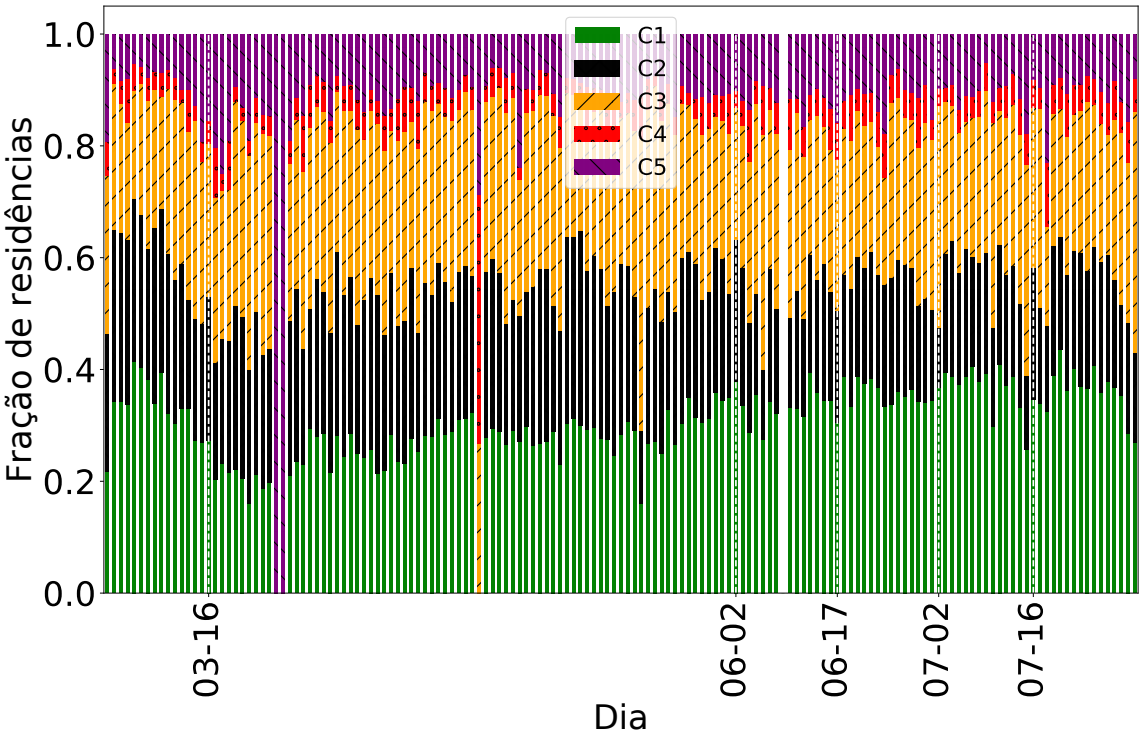


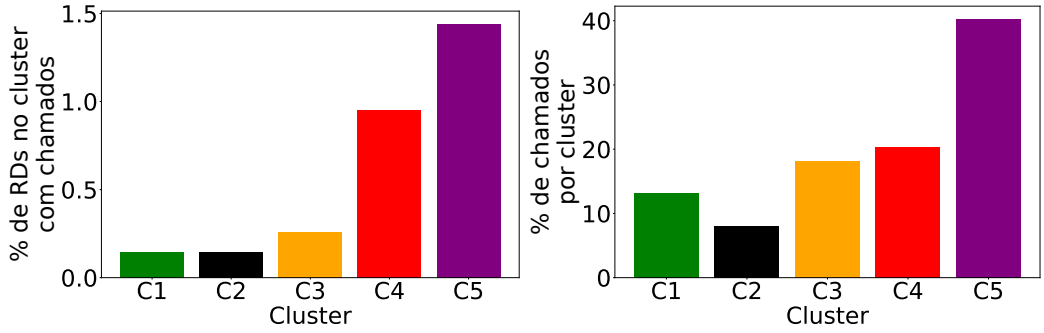
Figura 7.5: Resultados da clusterização entre os dias 01/03/2020 e 31/07/2020. O primeiro dia rotulado representa o início do período de confinamento enquanto os dias seguintes representam o início de diferentes fases de flexibilização da quarentena.

7.1.2 Resultados do método baseado em Detecção de Pontos de Mudança

Exemplificamos nesta seção a aplicação do método de detecção de pontos de mudança para localizar regiões da rede experimentando degradação de desempenho. Consideramos para a aplicação do método os resultados das medições de perda obtidos por 2485 residências distintas entre os dias 08/08/2018 e 23/09/2018. Filtramos amostras de perda com tráfego concorrente maior do que $\theta = 2.5$ Mbps e consideramos apenas séries temporais com ao menos $\psi = 1000$ amostras. Após a aplicação dos filtros, consideramos 76741 séries temporais, das quais 51041 séries temporais obtidas entre 08/08/2018 e 10/09/2018 são utilizadas para o *fit* do modelo e 25700, coletadas entre 11/09/2018 e 23/09/2018, são utilizadas para avaliação do método *online*.

Modelo de detecção de pontos de mudança

Diferentes codificações de símbolos e estruturas para a HMM foram comparadas utilizando o *log-likelihood* das sequências de valores observáveis. Baseados nesta avaliação, consideramos para a detecção de pontos de mudança uma HMM com $N = 4$ estados e $M = 22$ símbolos e limitamos as transições entre estados ocultos de forma a permitir apenas transições entre estados adjacentes (como mostrado na



(a) Percentual de RDs com chamado ao call center por cluster

(b) Percentual de chamados ao call center por cluster

Figura 7.6: Relação entre clusters obtidos e chamados ao call center.

Figura 7.7b). O resultado de uma medição de perda l_t é codificado como um símbolo observável O_t através da seguinte função:

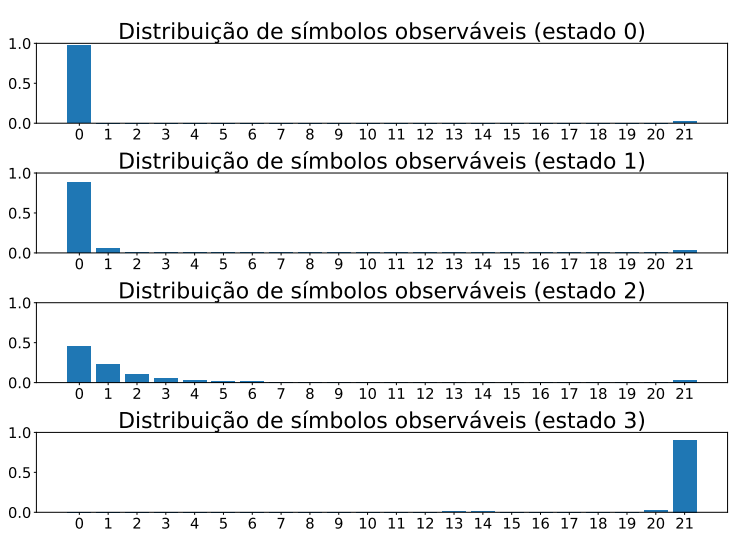
$$O_t = \begin{cases} l_t & \text{se } l_t \leq 19 \\ 20 & \text{se } l_t \geq 20 \end{cases}$$

Por fim, codificamos amostras faltantes utilizando o símbolo $\phi = 21$.

A cadeia oculta obtida após o treinamento é mostrada na Figura 7.7. É possível observar que o estado 2 está associado a uma maior probabilidade de perda de pacotes em comparação aos demais estados. Por outro lado, o estado 3 possui alta probabilidade de gerar amostras faltantes (símbolo 21), ou seja, este estado é associado a períodos de indisponibilidade da rede. A partir destes resultados, definimos o conjunto de estados associados a eventos na rede $\mathcal{R} = \{2, 3\}$. Aplicamos o filtro recursivo com uma janela de amostras de tamanho $\delta = 5$, enquanto o limiar para o tempo de permanência em $\mathcal{R}$ foi definido como $\gamma = 20$.

Resultados da correlação espaço-temporal

Utilizando o modelo de detecção de pontos de mudança obtido, exemplificamos o método de correlação espaço-temporal em uma região da rede com 42 clientes. A Figura 7.8 mostra os resultados da correlação espaço-temporal aplicada nos dias 08/08/2018 e 23/09/2018. A Figura 7.8a mostra que a grande maioria dos nós da topologia considerada apresentou degradação de desempenho no dia 08/08/2018. Durante o período considerado, foram realizadas pelo ISP diversas intervenções na região com o objetivo de melhorar a qualidade do serviço. Nossos resultados mostram que as mudanças realizadas melhoraram a qualidade da rede, uma vez que uma fração menor de nós é associada à degradação de desempenho no dia 23/09/2018 (Figura 7.8b). A melhoria no desempenho da rede detectada automaticamente pelo nosso método também pode ser observada a partir dos resultados das medições das



(a) Distribuição de símbolos observáveis

(b) Transição entre estados ocultos

Figura 7.7: Modelo de Markov oculto obtido após o treinamento

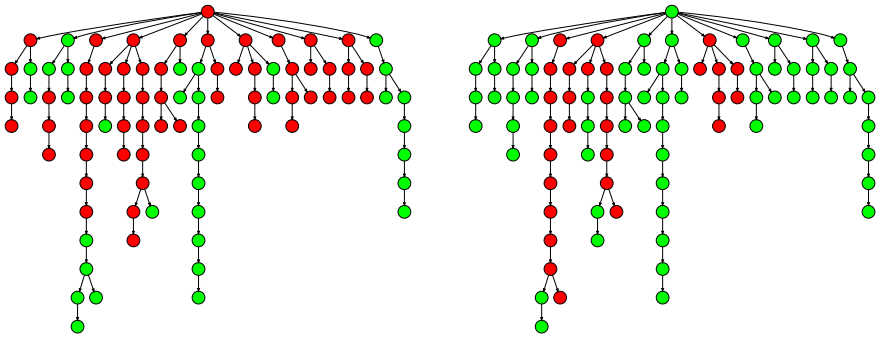
residências, como exemplifica a Figura 7.9. No entanto, uma quantidade considerável de nós da topologia ainda estava associada à degradação de desempenho no dia 23/09/2018, o que sugeria a necessidade de novas intervenções.

Para demonstrar a evolução temporal da qualidade do serviço prestado nesta região, avaliamos a fração de séries temporais associadas a estados do conjunto $\mathcal{R}$ em cada minuto, apresentada na Figura 7.10. Nota-se uma clara melhora no desempenho da rede para clientes associados a essa região no final do período considerado, com uma fração cada vez menor de séries temporais visitando estados em $\mathcal{R}$.

Avaliação do método *online*

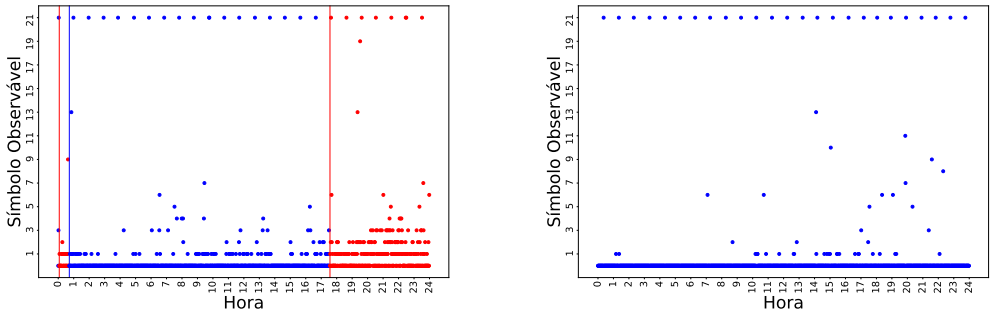
Em seguida, consideramos a aplicação do método proposto de maneira *online*. Nesta avaliação comparamos a sequência de estados ocultos $\mathbf{S}$ obtida com o algoritmo de Viterbi aplicado de maneira *offline* com a sequência de estados ocultos $\mathbf{S}^*$ obtida através da aplicação *online* do algoritmo de Viterbi (Seção 5.2). Consideramos nesta análise que a sequência de estados ocultos $\mathbf{S}$ representa o *ground truth*, ou seja, o objetivo do método *online* é obter uma sequência de estados ocultos $\mathbf{S}^*$ muito similar à sequência $\mathbf{S}$ obtida pelo método *offline*. Desta forma, utilizamos como métrica de avaliação a fração de estados ocultos s_t iguais nas sequências $\mathbf{S}$ e $\mathbf{S}^*$.

A Figura 7.11 mostra os resultados obtidos para diferentes tamanhos de janela. Observa-se uma maior acurácia à medida que o tamanho da janela aumenta, conforme esperado. Por outro lado, o aumento não é linear, sendo possível manter bom desempenho utilizando uma quantidade menor de amostras. Uma janela de 121



(a) Correlação espaço-temporal (08/08) (b) Correlação espaço-temporal (23/09)

Figura 7.8: Correlação espaço-temporal aplicada nos dias 08/08/2018 (esq.) e 23/09/2018 (dir.). Nós em vermelho experimentam degradação de performance.



(a) Série temporal antes da intervenção (08/08) (b) Série temporal após intervenção (23/09)

Figura 7.9: Mudança de comportamento da rede de usuário após intervenção na rede do ISP. Resultados em vermelho são associados a estados ocultos em $\mathcal{R}$

amostras, por exemplo, possui acurácia de aproximadamente 0,96 no conjunto de testes, enquanto uma janela de 61 amostras possui acurácia de aproximadamente 0,87. Os resultados mostram que o tamanho da janela deve ser escolhido considerando o *tradeoff* entre o tempo de detecção α e a acurácia do método.

Implementação de uma prova de conceito

Os métodos propostos neste trabalho possuem alta aplicabilidade em cenários reais. Por conta disso, a *startup* iniciada na incubadora da universidade financiou a implementação de uma prova de conceito do método *online* de detecção de pontos de mudança (Seção 5.2) com o objetivo de criar um produto para a detecção de eventos de rede em ISPs residenciais. Descrevemos brevemente nesta seção o funcionamento da ferramenta criada a partir do algoritmo proposto neste trabalho.

A Figura 7.12 mostra a interface da ferramenta que implementa o método de detecção e localização de degradação deste trabalho. A ferramenta utiliza marcado-

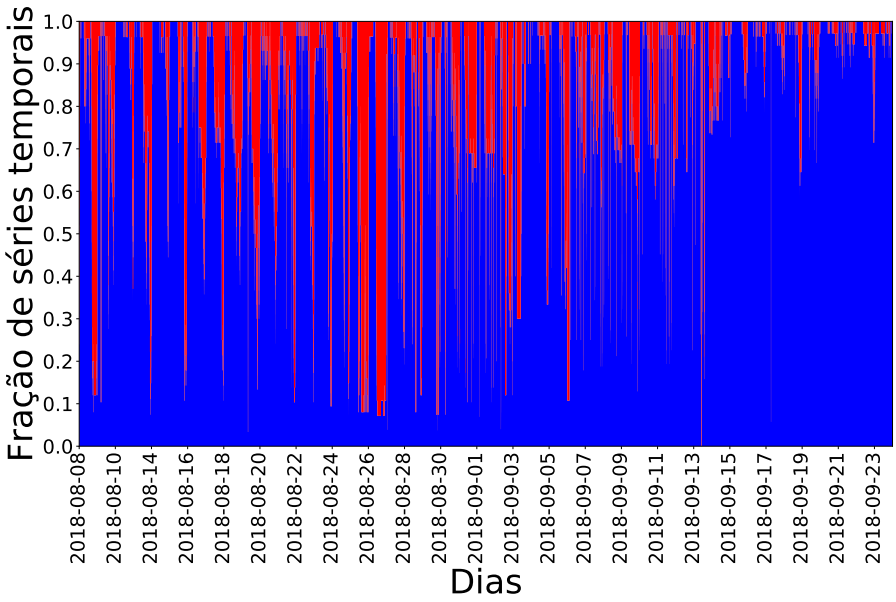


Figura 7.10: Fração de amostras das séries temporais associadas a estados em $\mathcal{R}$. Estados em $\mathcal{R}$ são representados pela cor vermelha.

res em um mapa para mostrar a localização de diferentes roteadores que realizam medições espalhados pela cidade de Nova Friburgo. Os resultados do algoritmo de detecção de pontos de mudança são utilizados para identificar quais medidores observam uma boa qualidade de serviço (marcador verde) e quais medidores observam degradação na QoS. O modelo proposto é utilizado para dividir a degradação de QoS observada em duas categorias: alta perda de pacotes (marcador vermelho) e indisponibilidade da rede (marcador roxo). Em seguida, utiliza-se o método de correlação espacial proposto para identificar quais regiões na rede do ISP observam degradação de desempenho.

A ferramenta é capaz de realizar notificações automáticas de problemas de rede detectados pelo nosso método através do envio de email ou de mensagem de Telegram para um conjunto configurável de destinatários. A detecção e localização de degradações na ferramenta são realizadas de maneira *online* a cada minuto a partir do resultado de medições de perda de milhares de roteadores, o que mostra que o método proposto é escalável e eficiente computacionalmente.

7.2 Método baseado em rótulos com ruído

Apresentamos a seguir os resultados obtidos pela aplicação do método de detecção e localização de degradação baseado em medições de QoS e rótulos de QoE com ruído. Nossa aplicação de exemplo considera os resultados de medições de perda de pacotes como métrica de QoS e os chamados técnicos realizados para o ISP parceiro como rótulos de QoE com ruído.

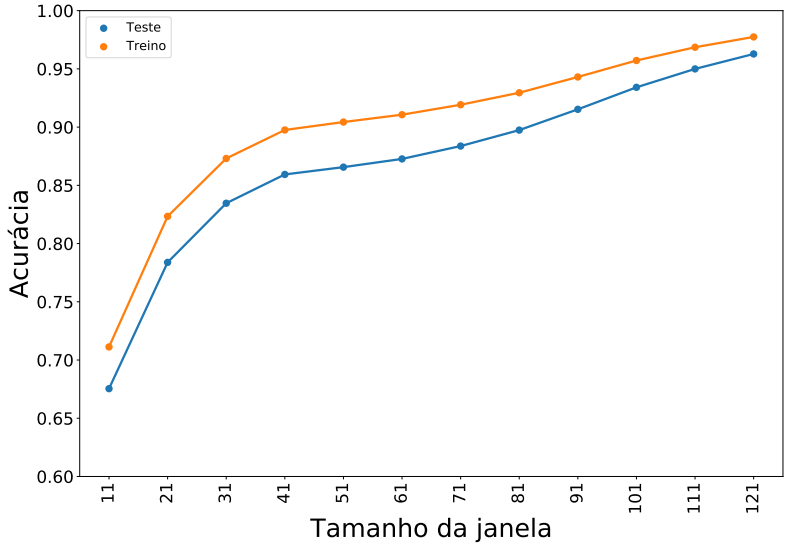


Figura 7.11: Taxa de acerto para aplicação de Viterbi *online*

Consideramos um dataset de 18 meses extraído a partir de medições de 6369 residências realizadas entre 16/01/2020 e 30/06/2021. Dividimos o dataset em três conjuntos: um conjunto de treinamento, coletado entre 16/01/2020 e 31/03/2020, um conjunto de validação, com dados coletados entre 01/04/2020 e 30/06/2020, e um conjunto de teste, com dados extraídos entre os dias 01/07/2020 e 30/06/2021. Durante a etapa de pré-processamento, filtramos em todos os conjuntos as amostras com tráfego concorrente maior que $c_{\max} = 2.5$ Mbps. Para o conjunto de treinamento, foram consideradas apenas séries temporais com ao menos $\psi = 1000$ amostras com o objetivo de reduzir o ruído durante o fit do modelo. Não filtramos séries temporais dos conjuntos de validação e teste com o objetivo de obter resultados mais próximos da aplicação em um cenário real. Após a aplicação do pré-processamento, obtivemos 282252 pares RD no conjunto de treinamento, 444341 pares RD no conjunto de validação e 1727574 pares RD no conjunto de teste.

Empregamos o conjunto de treinamento para realizar o fit do modelo. O conjunto de validação é utilizado para comparar a técnica baseada em rótulos com ruído com o método não supervisionado de detecção de pontos de mudança. Por fim, usamos o conjunto de teste para avaliar os resultados do modelo de correlação espaço-temporal e da técnica de ranqueamento de degradações.

7.2.1 Modelo de detecção baseado em rótulos com ruído

Relembramos que o modelo proposto correlaciona a perda de pacotes com os chamados ao call center através de uma variável latente Y representando a qualidade da rede, onde $Y = 0$ representa uma "qualidade boa" e $Y = 1$ representa uma "qua-

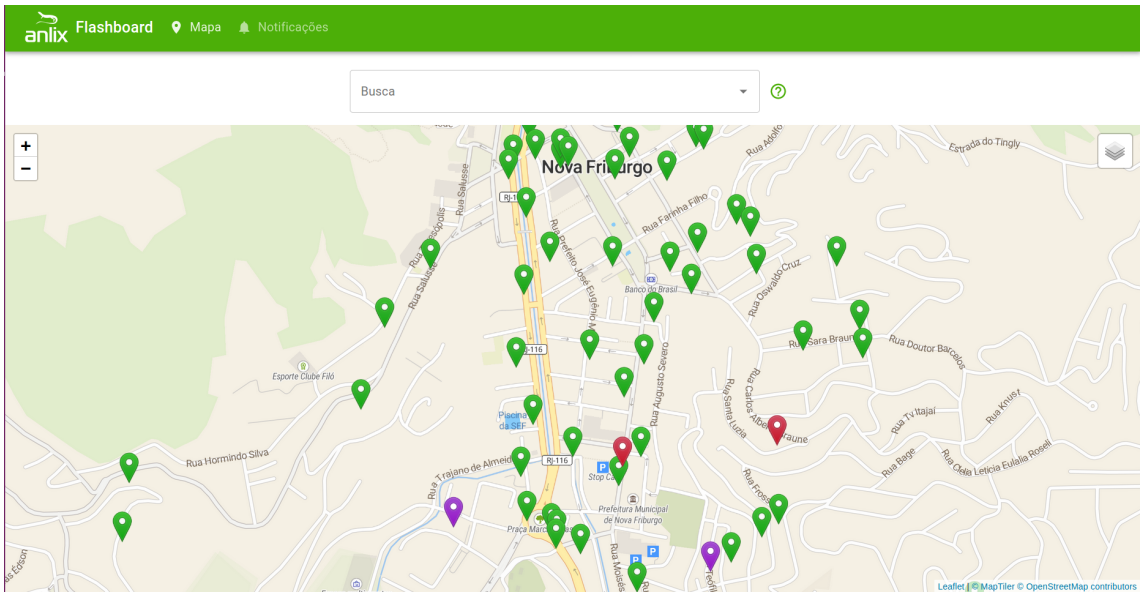


Figura 7.12: Implementação de prova de conceito. Marcadores indicam localização de roteadores e cor do marcador indica a QoS da rede identificada pelo método

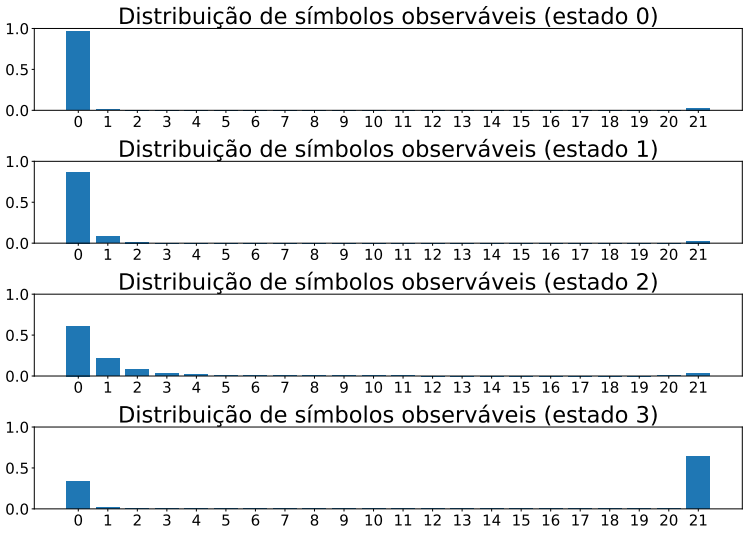
lidade ruim". Além disso, a distribuição de perda de pacotes condicionada ao valor da variável Y é dada por uma Cadeia de Markov Oculta (Seção 6.2). Consideramos cadeias ocultas com $S = 4$ estados e $|\mathcal{O}| = 22$ símbolos e limitamos as transições entre estados ocultos de forma a permitir apenas transições entre estados adjacentes, de maneira similar ao realizado para o modelo de detecção de pontos de mudança. Também aplicamos a mesma função para codificação de perda considerada anteriormente, isto é, o resultado de uma medição de perda l_t é codificado em um símbolo observável o usando a seguinte função:

$$o = \begin{cases} l_t & \text{se } l_t \leq 19 \\ 20 & \text{se } l_t \geq 20 \end{cases}$$

Por fim, codificamos amostras faltantes utilizando o símbolo 21.

As Figuras 7.13 e 7.14 mostram as distribuições de perda observáveis condicionadas ao valor da variável latente Y obtidas após o treinamento. É possível observar que as perdas são menos frequentes para o modelo associado a uma boa QoE (Figura 7.13). Além disso, períodos de indisponibilidade são mais prováveis no modelo associado a QoE ruim (Figura 7.14). Os resultados do modelo mostram que a maioria dos pares RD observa uma boa qualidade de rede (isto é, $P[Y = 0] > P[Y = 1]$). Note que a probabilidade de criação de um chamado ao call center é mais alta quando a qualidade da rede é pior, como esperado (isto é, $P[\tilde{Y} = 1|Y = 1] > P[\tilde{Y} = 1|Y = 0]$). Especificamente, a probabilidade de criação de chamado ao call center quando a qualidade da rede é considerada "ruim" é aproximadamente 4,5 vezes maior que a probabilidade obtida quando o modelo

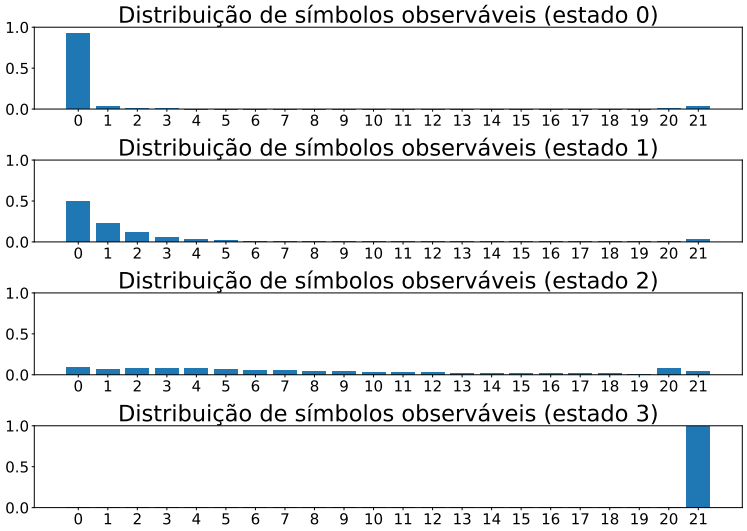
considera que a qualidade da rede é “boa”.



(a) Distribuição de símbolos observáveis

(b) Transição entre estados ocultos

Figura 7.13: Modelo obtido para $Y = 0$. $P[Y = 0] = 0.8208$ e $P[\tilde{Y} = 1|Y = 0] = 0.0026$



(a) Distribuição de símbolos observáveis

(b) Transição entre estados ocultos

Figura 7.14: Modelo obtido para $Y = 1$. $P[Y = 1] = 0.1792$ e $P[\tilde{Y} = 1|Y = 1] = 0.0116$

7.2.2 Resultados da correlação espaço-temporal

Devido à dificuldade de obtenção de rótulos identificando degradações de desempenho, adotamos uma avaliação quantitativa do método proposto baseada na probabilidade de criação de um chamado técnico ao call center. A análise parte da

suposição de que usuários que experimentam degradação no desempenho de sua rede têm maior probabilidade de abrir chamados técnicos para o call center do que aqueles que usufruem de boa QoS. Consequentemente, espera-se observar uma fração maior de clientes criando chamados técnicos ao call center quando há um evento de rede, comparada à fração de usuários abrindo chamados em períodos sem degradação de desempenho. Note que esta suposição não desconsidera o ruído inerente aos dados de chamados, pois a comparação contempla a possibilidade de abertura de chamados mesmo quando a rede de acesso apresenta boa QoS.

Um algoritmo de localização eficiente deve ser capaz de identificar quais regiões da rede estão experimentando degradação de desempenho. Desta forma, é esperado que se observe uma fração maior de residências abrindo chamados em regiões com QoS "ruim" identificadas por um algoritmo de localização eficiente quando comparada à fração de residências abrindo chamados em regiões com QoS "boa" detectadas pelo mesmo algoritmo.

Lembramos que o algoritmo de correlação espaço-temporal divide os nós da topologia da rede em dois conjuntos: $\mathcal{A}$, representando nós que experimentam degradação de desempenho, e $\bar{\mathcal{A}}$, representando nós com boa QoS de acordo com o método. Avaliamos os resultados do nosso método através da comparação entre a fração de residências conectadas a nós em $\mathcal{A}$ que criam chamados ao call center e a fração de residências que abrem chamados e estão conectadas a nós pertencentes a $\bar{\mathcal{A}}$.

Formalmente, definimos p_g (p_b) como a probabilidade de que uma residência do conjunto $\bar{\mathcal{A}}$ ($\mathcal{A}$) abra um chamado ao call center reportando um problema técnico em um determinado dia. De modo análogo, definimos $\hat{p}_g$ ($\hat{p}_b$) como o estimador dessa probabilidade. Espera-se que um algoritmo eficiente na detecção e localização de falhas apresente um valor de $\hat{p}_b$ superior ao de $\hat{p}_g$, isto é, esperamos que chamados ao call center sejam mais frequentes para pares RD pertencentes ao conjunto $\mathcal{A}$.

Aplicamos o nosso algoritmo de correlação espaço-temporal em cada dia dos nossos conjuntos de validação e teste. Desta forma, para cada dia d no conjunto de validação/teste, obtemos como resultado do algoritmo de correlação espaço-temporal um conjunto de nós experimentando degradação de desempenho no dia d, denotado por $\mathcal{A}_d$, e um conjunto de nós com boa QoS, denotado por $\overline{\mathcal{A}}_d$. Denotamos o número de residências conectadas a nós em $\mathcal{A}_d$ ($\overline{\mathcal{A}}_d$) como $n_{b,d}$ ($n_{g,d}$) e o número de residências com ao menos um chamado conectadas a nós em $\mathcal{A}_d$ ($\overline{\mathcal{A}}_d$) como $x_{b,d}$ ($x_{g,d}$).

Para avaliar nossos resultados, consideramos que p_g e p_b se mantêm constantes em diferentes dias no nosso dataset. Consequentemente, os resultados obtidos em dias distintos representam amostras de uma mesma distribuição binomial. Assim, é possível agregar os resultados obtidos durante um conjunto de dias $\mathcal{D}$ para obter os estimadores $\hat{p}_g$ e $\hat{p}_b$. Especificamente, estimadores de máxima verossimilhança para

p_g e p_b podem ser obtidos a partir de amostras em dias pertencentes a $\mathcal{D}$ através das seguintes equações:

$$\hat{p}_b = \frac{\sum_{\forall d \in \mathcal{D}} x_{b,d}}{\sum_{\forall d \in \mathcal{D}} n_{b,d}} \qquad \hat{p}_g = \frac{\sum_{\forall d \in \mathcal{D}} x_{g,d}}{\sum_{\forall d \in \mathcal{D}} n_{g,d}} \tag{7.1}$$

Para avaliar os resultados do algoritmo de correlação espaço-temporal, aplicamos o método de inferência bayesiana com uma distribuição *a priori* uniforme não-informativa, isto é, $\alpha_0 = 1$ e $\beta_0 = 1$. Consideramos um grau de confiança $\eta = 0.95$, um valor comumente usado na literatura. Definimos o valor para o limiar $\delta = 0.75$ com o objetivo de marcar como degradados apenas nós com ao menos 10 detecções, uma vez que $P[\phi_v > 0.75 | F_v = 10, n_v = 10] \approx 0.958$ e $P[\phi_v > 0.75 | F_v = 9, n_v = 9] \approx 0.944$.

Comparação com método não supervisionado

Utilizamos o conjunto de validação para comparar os resultados obtidos pelo método baseado em rótulos com ruído com o método não supervisionado de detecção de pontos de mudança. Ambos os modelos são treinados utilizando o mesmo conjunto de treinamento. Em seguida, utilizamos o conjunto de validação para obter a fração de chamados em regiões da rede com degradação de QoS detectada por cada método, isto é, comparamos o valor de $\hat{p}_b$ obtido por cada abordagem.

Inicialmente, avaliamos isoladamente o impacto dos dois modelos de detecção de degradação (modelo de detecção de pontos de mudança e o modelo baseado em rótulos com ruído) utilizando, em ambos, o mesmo algoritmo de correlação espaço-temporal baseado em votação majoritária. A Figura 7.15 mostra os resultados obtidos. É possível observar que o método que leva em consideração os chamados ao call center apresenta desempenho superior ao método não supervisionado, uma vez que regiões marcadas como potencialmente afetadas pelo método baseado em rótulos com ruído estão associadas a uma fração de chamados ao call center maior do que a obtida pelo método de detecção de pontos de mudança em todos os meses do conjunto de validação.

Em seguida, avaliamos isoladamente o impacto do método de correlação espaço-temporal. Para isso, comparamos os dois métodos (votação majoritária e inferência bayesiana) usando, em ambos os casos, os resultados de detecção obtidos pelo modelo baseado em rótulos com ruído. A Figura 7.16 mostra os resultados da comparação de $\hat{p}_b$ para cada método de correlação. Os resultados mostram que a correlação baseada em inferência bayesiana é mais eficiente que a correlação baseada em votação majoritária, uma vez que $\hat{p}_b$ é maior para o método de inferência bayesiana em todos os meses do conjunto de validação.

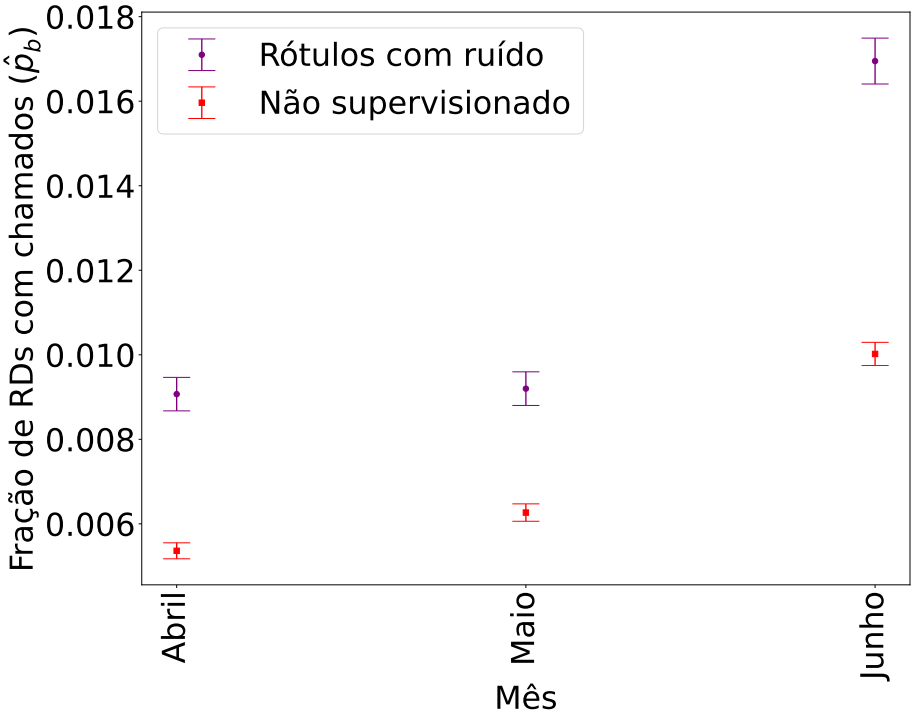


Figura 7.15: Comparação dos modelos de detecção utilizando votação majoritária (Intervalo de Confiança de 95%)

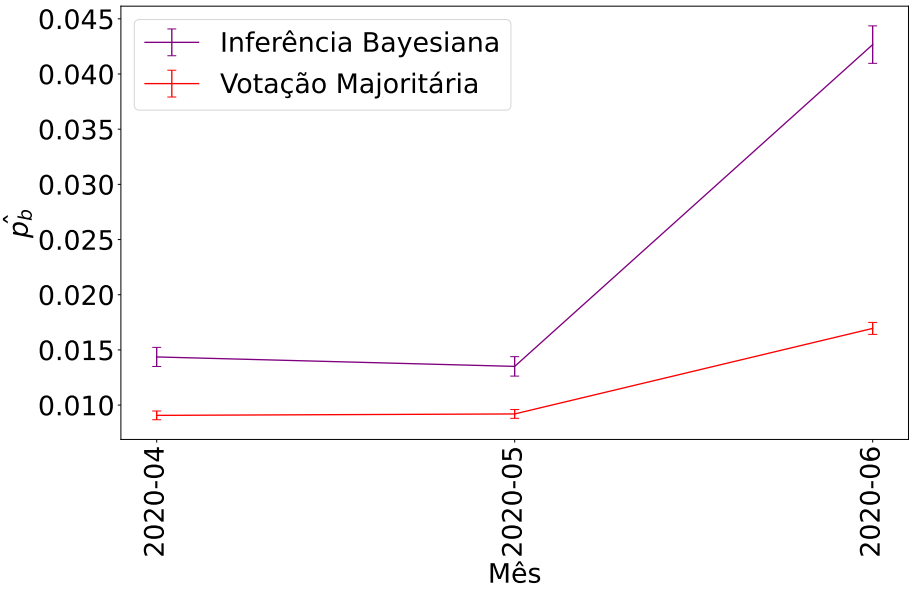


Figura 7.16: Comparação dos métodos de correlação espaço-temporal utilizando modelo baseado em rótulos com ruído (Intervalo de Confiança de 95%)

Resultados do conjunto de teste

Em seguida, estimamos $\hat{p}_g$ e $\hat{p}_b$ para cada mês do conjunto de teste (entre julho de 2020 e junho de 2021). A Figura 7.17 mostra os resultados obtidos. Para todos os meses considerados, o estimador $\hat{p}_b$, associado a RDs potencialmente afetados por um problema de rede, é uma ordem de grandeza maior que o estimador $\hat{p}_g$, associado a pares RD não afetados ($\bar{\mathcal{A}}$). Para comparar os estimadores $\hat{p}_b$ e $\hat{p}_g$, aplicamos um

teste de hipótese para comparação de proporções de duas populações [79] com um grau de significância $\alpha = 0.05$. Em todos os casos, podemos rejeitar a hipótese nula $p_b = p_g$ em favor da hipótese alternativa $p_b > p_g$. Desta forma, nossa evidência indica que a probabilidade de um usuário realizar um chamado é maior quando nosso método detecta uma piora na qualidade da rede.

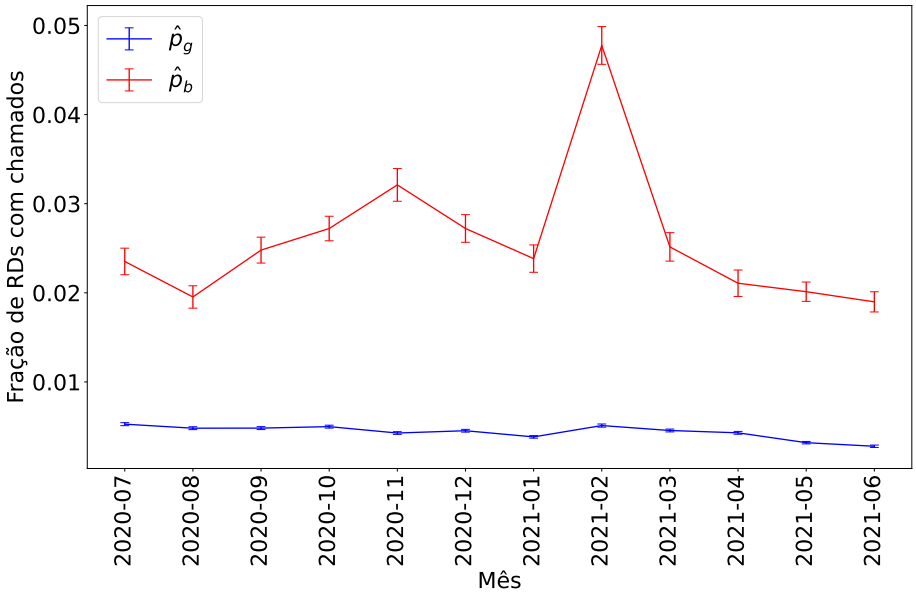


Figura 7.17: Comparação entre $\hat{p}_b$ e $\hat{p}_g$ no conjunto de teste (Intervalo de Confiança de 95%)

Uma métrica de interesse é a diferença entre as probabilidades p_b e p_g, que indica o quanto aumenta a probabilidade de realizar um chamado quando o algoritmo detecta uma degradação de desempenho. Usamos inferência bayesiana para estimar a distribuição de $(p_b - p_g)$. Modelamos o número de chamados em regiões com qualidade de rede boa/ruim utilizando uma distribuição binomial com parâmetro p_g (rede boa) / p_b (rede ruim). Adotamos para os 2 parâmetros uma distribuição *a priori* Beta(1, 1) (distribuição uniforme não-informativa). Dado o número de RDs com rede boa/ruim, a distribuição *a posteriori* de p_g e p_b também seguem uma distribuição Beta (distribuição *a priori* conjugada). Portanto, podemos estimar a distribuição *a posteriori* de $(p_b - p_g)$ a partir de amostras da distribuição *a posteriori* de p_b e p_g utilizando o método de Monte Carlo.

Mostramos os resultados obtidos para cada mês do conjunto de teste na Figura 7.18. É possível observar que a *Highest Posterior Density (HPD)* da distribuição de $(p_b - p_g)$ nunca inclui zero, isto é, existe uma forte evidência de que a probabilidade de haver um chamado aumenta quando nosso método detecta uma degradação na qualidade da rede. Além disso, o aumento na probabilidade pode ser de até 0.046, como observado em fevereiro de 2021.

É importante ressaltar que um valor de probabilidade da ordem de 10^{-2} não

é insignificante, pois pode representar centenas de chamados extras diários. Por exemplo, uma região de um ISP com 10000 usuários em $\mathcal{A}$ e $p_b - p_g = 0.014$ implica em uma média de 140 novos chamados em um dia.

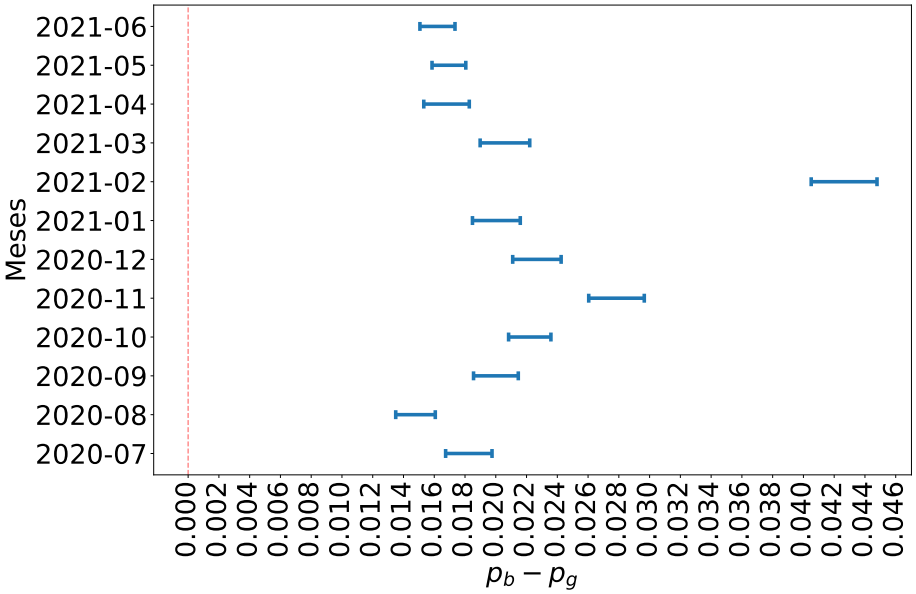


Figura 7.18: 95% HPD da distribuição da diferença entre p_b e p_g para cada mês

7.2.3 Resultados do método de ranqueamento de degradações

Apresentamos nesta seção a avaliação do desempenho do método de ranqueamento de degradações (Seção 6.5). Para avaliar o método de ranqueamento proposto, dividimos os problemas identificados pelo nosso método de localização em dois subconjuntos. Definimos regiões ranqueadas entre as 5 piores regiões com problemas localizados pelo nosso método de correlação como regiões com rank "alto", enquanto regiões com problemas ranqueadas fora do top 5 são definidas como rank "baixo". Avaliamos o desempenho do método de ranqueamento através da comparação entre a fração de residências abrindo chamados em regiões com rank "alto" e a fração de residências que criam chamados em regiões com rank "baixo".

Aplicamos o método de ranqueamento com um grau de confiança $\zeta = 0.95$ em cada dia d do nosso conjunto de teste. Lembramos que o conjunto de nós experimentando degradação de desempenho no dia d é denotado por $\mathcal{A}_d$. Para comparar o desempenho dos dois conjuntos de regiões ranqueadas (rank "alto" e rank "baixo") dividimos o conjunto $\mathcal{A}_d$ em dois subconjuntos: $\mathcal{A}_d^{\text{hi}}$, contendo todos os nós pertencentes a subárvores de rank "alto" e $\mathcal{A}_d^{\text{lo}}$, contendo nós pertencentes a subárvores de rank "baixo". Agregamos os resultados obtidos em diferentes dias de cada mês conforme descrito na Equação 7.1.

Comparamos o valor de $\hat{p}_b$ para nós em $\mathcal{A}_d^{\text{hi}}$ com o valor de $\hat{p}_b$ de nós em $\mathcal{A}_d^{\text{lo}}$ para cada mês do conjunto de teste, como mostrado na Figura 7.19. É possível observar que regiões com rank "alto" apresentam um valor superior para $\hat{p}_b$ na maioria dos meses do nosso conjunto de teste. Uma exceção pode ser observada em maio de 2021, quando o valor de $\hat{p}_b$ é significativamente maior para regiões fora do top 5 (isto é, não existe sobreposição dos intervalos de confiança). Neste mês, uma região *outlier* com muitas residências e uma fração alta de chamados criados foi ranqueada como a sexta pior região, o que explica o comportamento observado neste período.

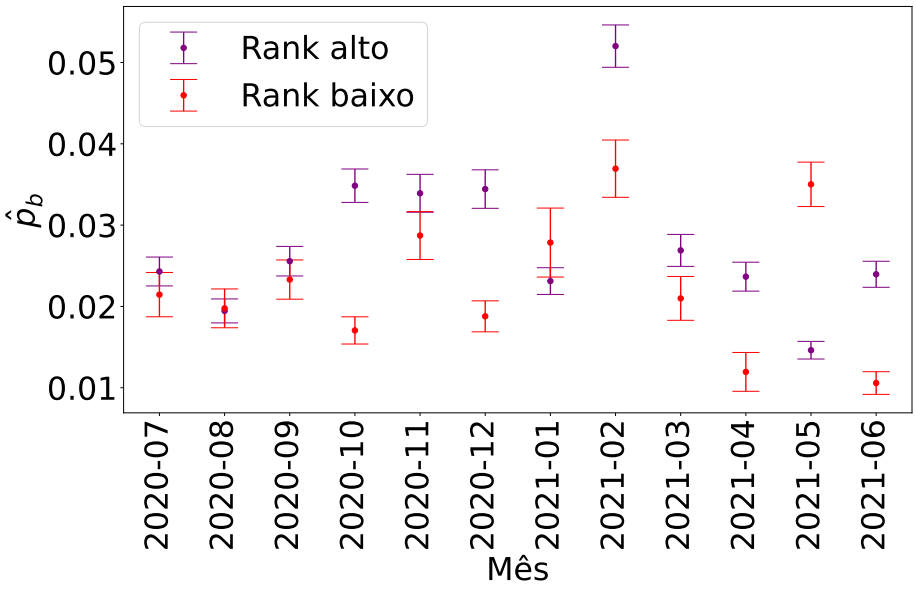


Figura 7.19: Avaliação de desempenho do método de ranqueamento

A evolução temporal dos resultados do ranqueamento também pode prover uma informação valiosa a respeito da natureza do problema observado. A Figura 7.20 mostra os resultados do método de ranqueamento aplicado aos dados de uma semana representativa no nosso conjunto de validação. O gráfico mostra que algumas regiões apresentam problemas crônicos afetando a QoE do usuário, como as regiões B e C, enquanto outras regiões apresentam problemas de maneira pontual. Esta informação pode auxiliar o ISP a priorizar esforços com o objetivo de corrigir problemas de desempenho crônicos durante o processo de *troubleshooting*.

7.2.4 Eventos de rede detectados

Apresentamos nessa seção alguns exemplos de degradações de desempenho detectadas pelo nosso método que tiveram sua causa raiz confirmada pelo ISP. Note que alguns dos exemplos mostrados também foram apresentados na aplicação do método não supervisionado baseado em decomposição tensorial (Seção 7.1.1).

A Figura 7.21a mostra a fração de residências associadas a uma QoE boa/ruim de acordo com nosso algoritmo de detecção para cada dia do mês de maio de 2020

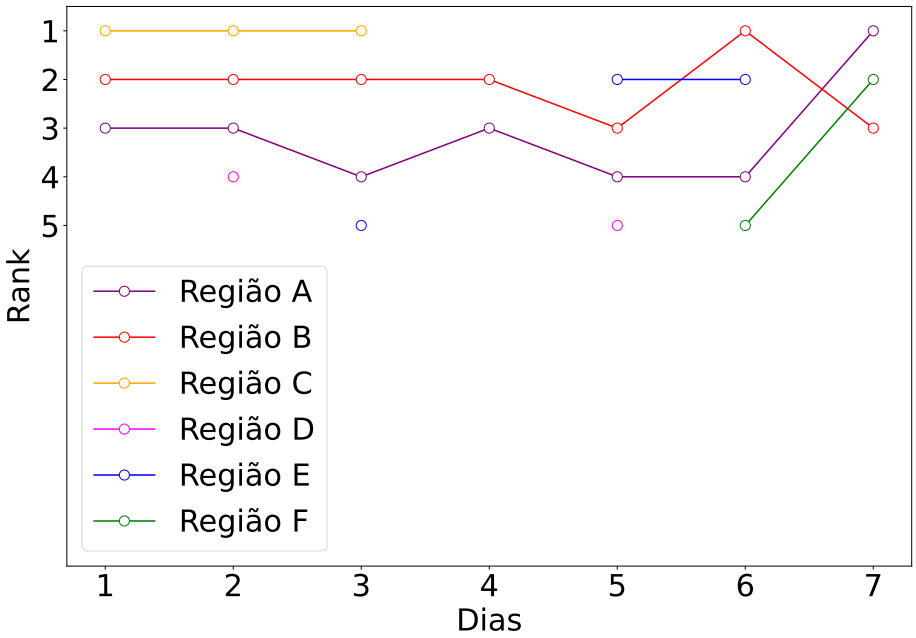


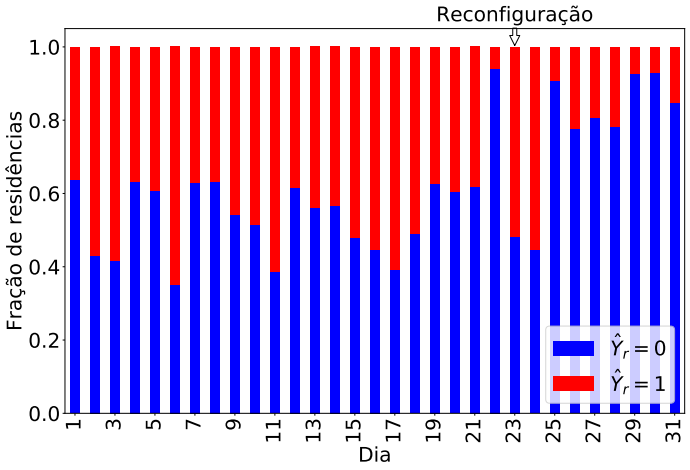
Figura 7.20: Evolução temporal do ranqueamento durante uma semana filtrando regiões com ao menos 2 dias no top 5

em uma determinada região da rede. Os resultados mostram que o nosso método frequentemente detectava uma qualidade ruim para um número considerável de residências até o dia 24 (entre 40% e 60% das residências aproximadamente). O ISP parceiro nos informou que uma reconfiguração na rede tinha sido iniciada no dia 23 para melhorar a QoS. Nosso método identificou o impacto da intervenção do ISP: é possível observar que o percentual de residências com QoE ruim caiu para a faixa de 10% a 20%, aproximadamente, após o dia 25.

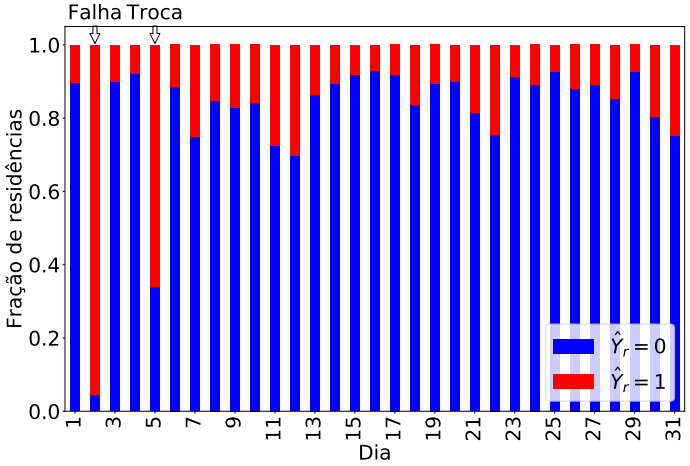
A Figura 7.21b apresenta os resultados obtidos pelo nosso método em outra região da rede durante o mês de agosto de 2020. É possível observar que a fração de residências com boa QoE estimada pelo algoritmo é em geral alta na maior parte do tempo (acima de 80%). No entanto, nosso algoritmo detectou degradação na QoE para uma fração significativa de residências nos dias 2 e 5. Uma falha no equipamento foi reportada pelo ISP no dia 2, tornando a rede indisponível por algumas horas. Este equipamento foi trocado no dia 5, e a rede também apresentou indisponibilidade durante o período de manutenção.

Por fim, a Figura 7.21c apresenta os resultados do método para uma região da rede em junho de 2021. Uma falha de equipamento ocorreu no dia 18, interrompendo o acesso à rede de múltiplas residências. No entanto, o gráfico mostra que o nosso método identificou uma grande quantidade de residências detectando degradação na qualidade da rede desde o dia 7. Múltiplas residências observaram altas perdas (ao menos 20%) em vários minutos entre os dias 7 e 18, o que indica que o equipamento começou a falhar muito antes de sua troca. Este exemplo mostra como a técnica proposta pode auxiliar a manutenção proativa da rede: o ISP pode utilizar nosso

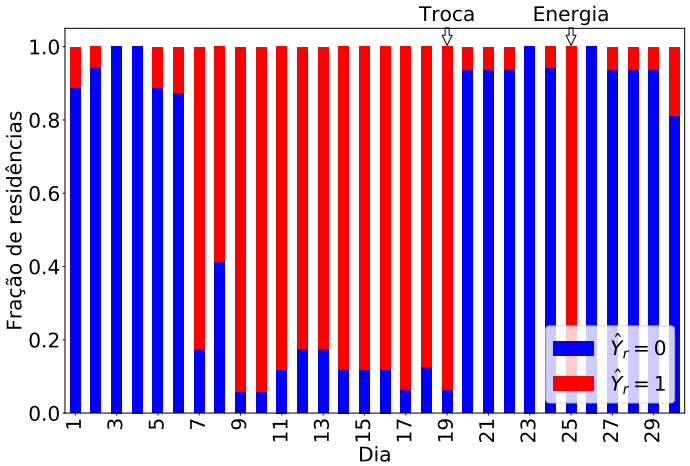
método para identificar falhas na rede assim que elas surgem, minimizando o seu impacto na qualidade da experiência dos usuários. Note que uma falha não relacionada aconteceu no dia 25, quando um problema elétrico causou indisponibilidade na rede de múltiplas residências.



(a) Reconfiguração da rede iniciada no dia 23



(b) Falha de equipamento no dia 2. Equipamento foi substituído no dia 5



(c) Equipamento substituído no dia 19. Método detecta degradação na QoS desde o dia 7

Figura 7.21: Exemplos de degradações reais detectadas pelo método

Capítulo 8

Conclusão

Propomos neste trabalho novas abordagens para detecção e localização de degradação de desempenho de rede capazes de lidar com a falta de rótulos confiáveis que identifiquem esses eventos. Os métodos propostos utilizam um framework baseado nos resultados da aplicação de modelos de aprendizado de máquina para extrair padrões de degradação de forma automática. Exemplificamos as técnicas propostas utilizando um dataset real incluindo resultados de medições de QoS, a topologia da rede e os chamados técnicos realizados ao *call center*.

Os métodos desenvolvidos consideram dois cenários distintos. Para o cenário em que nenhum rótulo está disponível, propomos duas técnicas baseadas em modelos não supervisionados. A primeira técnica apresentada se baseia na clusterização dos residuais obtidos através da decomposição tensorial de séries temporais multivariadas de latência e perda para identificar diferentes tipos de degradação, como congestionamentos e períodos de indisponibilidade. Também propomos uma segunda técnica não supervisionada que utiliza séries temporais univariadas de perda como entrada para um método de detecção de pontos de mudança estatística baseado em Cadeias de Markov Ocultas capaz de detectar e localizar equipamentos que experimentam degradação de desempenho.

Consideramos também o cenário em que chamados técnicos realizados ao call center estão disponíveis para a construção de modelos. Chamados ao call center apresentam ruído inerente ao comportamento dos usuários, o que torna inadequada a utilização de métodos supervisionados. Desta forma, propomos um novo modelo de detecção de degradação capaz de lidar com o ruído de maneira apropriada. Desenvolvemos um método de aprendizado de máquina estatístico capaz de identificar o estado latente da rede a partir de métricas de QoS e de rótulos de QoE com ruído. O método correlaciona a existência de chamados ao *call center* com o resultado de medições de perda de pacotes através de um modelo probabilístico otimizado pelo algoritmo de Expectation-Maximization. Elaboramos uma técnica de correlação espaço-temporal dos resultados para localizar degradações de desempenho através

de um método baseado em inferência bayesiana. Por fim, desenvolvemos um novo método de ranqueamento que ordena as regiões da rede de acordo com o grau de confiança na detecção, com o objetivo de auxiliar o ISP a priorizar os esforços de sua equipe operacional durante o processo de *troubleshooting*.

Demonstramos a eficiência dos métodos propostos utilizando dados reais coletados por milhares de medidores implementados em roteadores residenciais por dezenas de meses. Mostramos que os nossos métodos são capazes de detectar eventos reais de degradação confirmados pelo ISP. Além disso, mostramos a aplicabilidade dos nossos métodos em cenários reais a partir da implementação de uma prova de conceito capaz de detectar degradações de maneira *online*.

As técnicas propostas neste trabalho têm como foco detectar e localizar degradação no desempenho da rede de acesso de provedores residenciais. Trabalhos futuros incluem a avaliação de outros métodos de correlação espaço-temporal capazes de considerar, durante o processo de agregação, a probabilidade de degradação do desempenho da rede extraída a partir dos resultados dos detectores. Também é possível considerar a aplicação do método de detecção baseado em rótulos com ruído para correlacionar outras métricas de QoS e de QoE, como a correlação da vazão e da latência com a avaliação subjetiva da qualidade de experiência em aplicações de vídeo. Por fim, a obtenção de um conjunto robusto de rótulos identificando degradações de desempenho possibilitaria a utilização de um método semissupervisionado combinando os dois tipos de rótulos disponíveis (com e sem ruído).

Referências Bibliográficas

[1] CHANDOLA, V., BANERJEE, A., KUMAR, V. "Anomaly detection: A survey", *ACM computing surveys (CSUR)*, v. 41, n. 3, pp. 1–58, 2009.

[2] LAKHINA, A., CROVELLA, M., DIOT, C. "Mining anomalies using traffic feature distributions". In: *ACM computer communication review*, v. 35, pp. 217–228, 2005.

[3] SILVEIRA, F., DIOT, C., TAFT, N., et al. "ASTUTE: Detecting a different class of traffic anomalies", *ACM SIGCOMM Computer Communication Review*, v. 40, n. 4, pp. 267–278, 2010.

[4] SILVEIRA, F., DIOT, C. "URCA: Pulling out anomalies by their root causes". In: *2010 Proceedings IEEE INFOCOM*, pp. 1–9. IEEE, 2010.

[5] DUFFIELD, N. "Network tomography of binary network performance characteristics", *IEEE Transactions on Information Theory*, v. 52, n. 12, pp. 5373–5388, 2006.

[6] DUFFIELD, N., PRESTI, F. L., PAXSON, V., et al. "Network loss tomography using striped unicast probes", *IEEE/ACM Transactions on Networking*, v. 14, n. 4, pp. 697–710, 2006.

[7] GHITA, D., NGUYEN, H., KURANT, M., et al. "Netscope: Practical network loss tomography". In: *2010 Proceedings IEEE INFOCOM*, pp. 1–9. IEEE, 2010.

[8] XIE, K., LI, X., WANG, X., et al. "Graph based tensor recovery for accurate internet anomaly detection". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1502–1510. IEEE, 2018.

[9] NATARAJAN, N., DHILLON, I. S., RAVIKUMAR, P. K., et al. "Learning with noisy labels", *Advances in neural information processing systems*, v. 26, pp. 1196–1204, 2013.

[10] STREIT, A. G., SANTOS, G. H., LEAO, R. M., et al. "Identificação de Anomalias em Redes de Dados baseada em Decomposição Tensorial". In: *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 952–965. SBC, 2020.

[11] STREIT, A., SANTOS, G., LEÃO, R., et al. "Network Anomaly Detection based on Tensor Decomposition". In: *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*, pp. 1–8. IEEE, 2020.

[12] STREIT, A., SANTOS, G. H., LEÃO, R. M., et al. "Network anomaly detection based on tensor decomposition", *Computer Networks*, v. 200, pp. 108503, 2021.

[13] SANTOS, G. H., MENDONÇA, G., DE SOUZA, E., et al. "Análise não supervisionada para inferência de qualidade de experiência de usuários residenciais". In: *SBRC 2019*, pp. 958–971, 2019.

[14] SANTOS, G. H., MENDONÇA, G., LEÃO, R. M., et al. "Detecção de anomalias em redes baseada em medições de QoS e rótulos de QoE com ruído". In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 98–111. SBC, 2022.

[15] LAKHINA, A., CROVELLA, M., DIOT, C. "Diagnosing network-wide traffic anomalies". In: *Comp. Comm. Review*, v. 34, pp. 219–230, 2004.

[16] YAN, H., FLAVEL, A., GE, Z., et al. "Argus: End-to-end service anomaly detection and localization from an ISP's point of view". In: *INFOCOM, 2012 Proceedings IEEE*, pp. 2756–2760, 2012.

[17] BARFORD, P., DUFFIELD, N., RON, A., et al. "Network performance anomaly detection and localization". In: *IEEE INFOCOM 2009*, pp. 1377–1385. IEEE, 2009.

[18] ZHAO, Z., CERF, S., BIRKE, R., et al. "Robust anomaly detection on unreliable data". In: *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 630–637. IEEE, 2019.

[19] MARUHASHI, K., GUO, F., FALOUTSOS, C. "Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis". In: *Advances in Social Networks Analysis & Mining*, pp. 203–210, 2011.

[20] KASAI, H., KELLERER, W., KLEINSTEUBER, M. "Network volume anomaly detection and identification in large-scale networks based on online

time-structured traffic tensor tracking", *IEEE Transactions on Network and Service Management*, v. 13, n. 3, pp. 636–650, 2016.

[21] LI, X., XIE, K., WANG, X., et al. "Online Internet anomaly detection with high accuracy: A fast tensor factorization solution". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1900–1908. IEEE, 2019.

[22] HERODOTOU, H., DING, B., BALAKRISHNAN, S., et al. "Scalable near real-time failure localization of data center networks". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1689–1698, 2014.

[23] PADMANABHAN, V. N., QIU, L., WANG, H. J. "Server-based inference of Internet link lossiness". In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, v. 1, pp. 145–155. IEEE, 2003.

[24] CHOFFNES, D. R., BUSTAMANTE, F. E., GE, Z. "Crowdsourcing service-level network event monitoring". In: *Proceedings of the ACM SIGCOMM 2010 Conference*, pp. 387–398, 2010.

[25] SUNDARESAN, S., DE DONATO, W., N.FEAMSTER, et al. "Broadband Internet Performance: A View From the Gateway". In: *Proceedings of the ACM SIGCOMM 2011*, 2011.

[26] BROCKWELL, P. J., DAVIS, R. A. *Time series: theory and methods*. Springer science & business media, 2009.

[27] XIAO, J., LI, Q., ZHAO, D., et al. "Themis: A passive-active hybrid framework with in-network intelligence for lightweight failure localization", *Computer Networks*, v. 255, pp. 110836, 2024.

[28] TAN, C., JIN, Z., GUO, C., et al. "Netbouncer: Active device and link failure localization in data center networks". In: *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pp. 599–614, 2019.

[29] GUO, C., YUAN, L., XIANG, D., et al. "Pingmesh: A large-scale system for data center network latency measurement and analysis". In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 139–152, 2015.

[30] PENG, Y., YANG, J., WU, C., et al. "deTector: a topology-aware monitoring system for data center networks". In: *2017 {USENIX} Annual Technical Conference ({USENIX}{ATC} 17)*, pp. 55–68, 2017.

[31] DHAMDHERE, A., TEIXEIRA, R., DOVROLIS, C., et al. "NetDiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data". In: *Proceedings of the 2007 ACM CoNEXT conference*, pp. 1–12, 2007.

[32] GENG, Y., LIU, S., YIN, Z., et al. "{SIMON}: A Simple and Scalable Method for Sensing, Inference and Measurement in Data Center Networks". In: *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pp. 549–564, 2019.

[33] GONG, F., RAGHUNATHAN, D., GUPTA, A., et al. "Zoom2net: Constrained network telemetry imputation". In: *Proceedings of the ACM SIGCOMM 2024 Conference*, pp. 764–777, 2024.

[34] ZHENG, H., HUANG, C., HAN, X., et al. "μMon: Empowering Microsecond-level Network Monitoring with Wavelets". In: *Proceedings of the ACM SIGCOMM 2024 Conference*, pp. 274–290, 2024.

[35] CÁCERES, R., DUFFIELD, N. G., HOROWITZ, J., et al. "Multicast-based inference of network-internal loss characteristics", *IEEE Transactions on Information theory*, v. 45, n. 7, pp. 2462–2480, 1999.

[36] PRESTI, F. L., DUFFIELD, N. G., HOROWITZ, J., et al. "Multicast-based inference of network-internal delay distributions", *IEEE/ACM Transactions On Networking*, v. 10, n. 6, pp. 761–775, 2002.

[37] NGUYEN, H. X., THIRAN, P. "The boolean solution to the congested IP link location problem: Theory and practice". In: *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pp. 2117–2125. IEEE, 2007.

[38] CUNHA, Í., TEIXEIRA, R., FEAMSTER, N., et al. "Measurement methods for fast and accurate blackhole identification with binary tomography". In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pp. 254–266, 2009.

[39] GHITA, D., KARAKUS, C., ARGYRAKI, K., et al. "Shifting network tomography toward a practical goal". In: *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pp. 1–12, 2011.

[40] SALAMATIAN, L., VERMEULEN, K., CHOFFNES, D., et al. "HERMES: Repurposing User-Driven Speed Tests to Monitor the Internet". In: *Proceedings of the ACM SIGCOMM 2026 Conference*, pp. 966–989, 2026.

[41] KANDULA, S., KATABI, D., VASSEUR, J.-P. "Shrink: A tool for failure diagnosis in IP networks". In: *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pp. 173–178, 2005.

[42] BAHL, P., CHANDRA, R., GREENBERG, A., et al. "Towards highly reliable enterprise network services via inference of multi-level dependencies", *ACM SIGCOMM Computer Communication Review*, v. 37, n. 4, pp. 13–24, 2007.

[43] KOMPELLA, R. R., YATES, J., GREENBERG, A., et al. "Fault localization via risk modeling", *IEEE Transactions on Dependable and Secure Computing*, v. 7, n. 4, pp. 396–409, 2009.

[44] MAHIMKAR, A. A., GE, Z., SHAIKH, A., et al. "Towards automated performance diagnosis in a large IPTV network", *ACM SIGCOMM Computer Communication Review*, v. 39, n. 4, pp. 231–242, 2009.

[45] JIN, Y., DUFFIELD, N., GERBER, A., et al. "Nevermind, the problem is already fixed: proactively detecting and troubleshooting customer dsl problems". In: *Proceedings of the 6th International COnference*, pp. 1–12, 2010.

[46] SONG, H. H., GE, Z., MAHIMKAR, A., et al. "Q-score: Proactive service quality assessment in a large IPTV system". In: *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pp. 195–208, 2011.

[47] PAN, L., ZHANG, J., LEE, P. P., et al. "An intelligent customer care assistant system for large-scale cellular network diagnosis". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge discovery and data mining*, pp. 1951–1959, 2017.

[48] DEB, S., GE, Z., ISUKAPALLI, S., et al. "Aesop: Automatic policy learning for predicting and mitigating network service impairments". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1783–1792, 2017.

[49] HU, J., ZHOU, Z., YANG, X., et al. "CableMon: Improving the reliability of cable broadband networks via proactive network maintenance". In: *17th*

{*USENIX*} *Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pp. 619–632, 2020.

[50] FRÉNAY, B., VERLEYSEN, M. "Classification in the presence of label noise: a survey", *IEEE transactions on neural networks and learning systems*, v. 25, n. 5, pp. 845–869, 2013.

[51] LAWRENCE, N., SCHÖLKOPF, B. "Estimating a kernel fisher discriminant in the presence of label noise". In: *18th International Conference on Machine Learning (ICML 2001)*, pp. 306–306. Morgan Kaufmann, 2001.

[52] FRÉNAY, B., LANNOY, G. D., VERLEYSEN, M. "Label noise-tolerant hidden Markov models for segmentation: application to ECGs". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 455–470. Springer, 2011.

[53] KOLDA, T. G., BADER, B. W. "Tensor decompositions and applications", *SIAM review*, v. 51, n. 3, pp. 455–500, 2009.

[54] BRO, R. "PARAFAC. Tutorial and applications", *Chemometrics and intelligent laboratory systems*, v. 38, n. 2, pp. 149–171, 1997.

[55] FANAEE-T, H., GAMA, J. "Tensor-based anomaly detection: An interdisciplinary survey", *Knowledge-Based Systems*, v. 98, pp. 130–147, 2016.

[56] HARSHMAN, R. A. ""How can I know if it's real?"A catalogue of diagnostics for use with three-mode factor analysis", *Research methods for multimode data analysis*, pp. 566–591, 1984.

[57] LORENZO-SEVA, U., TEN BERGE, J. M. "Tucker's congruence coefficient as a meaningful index of factor similarity", *Methodology*, v. 2, n. 2, pp. 57–64, 2006.

[58] SALAMATIAN, K., VATON, S. "Hidden Markov Modeling for network communication channels". In: *Proceedings of the ACM SIGMETRICS*, pp. 92–101, 2001.

[59] WEI, W., WANG, B., TOWSLEY, D. "Continuous-time hidden Markov models for network performance evaluation", *Performance Evaluation*, v. 49, n. 1-4, pp. 129–146, 2002.

[60] SILVEIRA, F., DE SOUZA E SILVA, E. "Predicting packet loss statistics with hidden Markov models for FEC control", *Computer Networks*, v. 56, pp. 628–641, 2012.

[61] RABINER, L. R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, 1989. ISSN: 15582256. doi: 10.1109/5.18626.

[62] DE SOUZA E SILVA, E., LEÃO, R. M. M., MUNTZ., R. R. "Performance Evaluation with Hidden Markov Models". In: *Performance Evaluation of Computer and Communication Systems. Milestones and Future Challenges*, pp. 112–128, 2011.

[63] DA SILVA, A. P. C., VARELA, M., DE SOUZA E SILVA, E., et al. "Quality assessment of interactive voice applications", *Computer Networks*, v. 52, n. 6, pp. 1179–1192, April 2008 2008.

[64] CHARONYKTAKIS, P., PLAKIA, M., TSAMARDINOS, I., et al. "On User-Centric Modular QoE Prediction for VoIP Based on Machine-Learning Algorithms", *IEEE Transactions on Mobile Computing*, v. 15, pp. 1443–1456, 2016.

[65] RINGBERG, H., SOULE, A., REXFORD, J., et al. "Sensitivity of PCA for traffic anomaly detection". In: *ACM SIGMETRICS Performance Evaluation Review*, v. 35, pp. 109–120, 2007.

[66] SIDIROPOULOS, N. D., DE LATHAUWER, L., FU, X., et al. "Tensor decomposition for signal processing and machine learning", *IEEE Transactions on Signal Processing*, v. 65, n. 13, pp. 3551–3582, 2017.

[67] STREIT, A., LEÃO, R., DE SOUZA E SILVA, E., et al. "Descobrindo perfis de tráfego de usuários: uma abordagem não supervisionada". In: *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 169–182, Porto Alegre, RS, Brasil, 2019. SBC. doi: 10.5753/sbrc.2019.7358. Disponível em: <`https://sol.sbc.org.br/index.php/sbrc/article/view/7358`>.

[68] YU, S.-Z., KOBAYASHI, H. "A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking", *Signal Processing*, v. 83, n. 2, pp. 235–250, 2003.

[69] AMINIKHANGHAHI, S., COOK, D. J. "A survey of methods for time series change point detection", *Knowledge and information systems*, v. 51, n. 2, pp. 339–367, 2017.

[70] XIMENES, D., MENDONÇA, G., SANTOS, G. H. A., et al. "O Problema de Deteção e Localização de Eventos em Séries Temporais Aplicado a

Redes de Computadores", *Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance CSBC)*, v. 17, n. 1/2018, 2018. ISSN: 2595-6167. Disponível em: <http://portaldeconteudo.sbc.org.br/index.php/wperformance/article/view/3323>.

[71] DE ALMEIDA, C. M., LEÃO, R. M., DE SOUZA E SILVA, E. "Inferring change points in unlabelled time series data collected from the network diagnosis tool", *Annals of Telecommunications*, v. 80, n. 9, pp. 775–792, 2025.

[72] KEHAGIAS, A. "A hidden Markov model segmentation procedure for hydrological and environmental time series", *Stochastic Environmental Research and Risk Assessment*, v. 18, n. 2, pp. 117–130, 2004. ISSN: 1436-3259. doi: 10.1007/s00477-003-0145-5.

[73] LUONG, T. M., ROZENHOLC, Y., NUEL, G. "Fast estimation of posterior probabilities in change-point analysis through a constrained hidden Markov model", *Computational Statistics and Data Analysis*, v. 68, pp. 129 – 140, 2013. ISSN: 0167-9473. doi: http://dx.doi.org/10.1016/j.csda.2013.06.020.

[74] MONTAÑEZ, G. D., AMIZADEH, S., LAPTEV, N. "Inertial Hidden Markov Models: Modeling Change in Multivariate Time Series". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 1819–1825. AAAI Press, 2015. ISBN: 0-262-51129-0.

[75] HU, J., ZHOU, Z., YANG, X. "Characterizing Physical-Layer Transmission Errors in Cable Broadband Networks". In: *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pp. 845–859, 2022.

[76] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg, Springer-Verlag, 2006. ISBN: 0387310738.

[77] DEMPSTER, A. P., LAIRD, N. M., RUBIN, D. B. "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society: Series B (Methodological)*, v. 39, n. 1, pp. 1–22, 1977.

[78] GELMAN, A., CARLIN, J. B., STERN, H. S., et al. *Bayesian data analysis*. CRC press, 2013.

[79] MONTGOMERY, D. C., RUNGER, G. C. *Applied statistics and probability for engineers*. John Wiley & Sons, 2010.

Apêndice A

Resultados Auxiliares

A.1 Algoritmo de Expectation-Maximization

Descrevemos nesta seção os passos necessários para derivar as equações mostradas na Seção 6.2.2. O primeiro passo da derivação é definir a distribuição conjunta das variáveis aleatórias de interesse. Utilizando o PGM (Figura 6.1) é fácil ver que:

$$P[\tilde{\mathbf{Y}}, \mathbf{Y}, \mathbf{O}, \mathbf{q}] = P[\mathbf{Y}]P[\tilde{\mathbf{Y}}|\mathbf{Y}]P[\mathbf{O}, \mathbf{q}|\mathbf{Y}] \tag{A.1}$$

Onde $\mathbf{Y} = (Y_1, Y_2, ..., Y_R)$, $\tilde{\mathbf{Y}} = (\tilde{Y_1}, \tilde{Y_2}, ..., \tilde{Y_R})$, $\mathbf{O} = (\mathbf{o}^1, \mathbf{o}^2, ..., \mathbf{o}^R)$ e $\mathbf{q} = (\mathbf{q}^1, \mathbf{q}^2, ..., \mathbf{q}^R)$.

Em seguida, definimos a *log-likelihood* da distribuição conjunta. Através da aplicação do logaritmo e do uso de álgebra simples, obtemos:

$$\begin{aligned}
\log P[\tilde{\mathbf{Y}}, \mathbf{Y}, \mathbf{O}, \mathbf{q}] = &\sum_{r=1}^{R}\sum_{l=0}^{1} \mathbb{1}(Y_r = l) \log \rho_l + \\
&\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{m=0}^{1} \mathbb{1}(Y_r = l)\mathbb{1}(\tilde{Y_r} = m) \log \varepsilon_{m|l} + \\
&\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{k=0}^{S-1} \mathbb{1}(Y_r = l)\mathbb{1}(q_1^r = k) \log \pi_k^{(l)} + \\
&\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{t=1}^{T-1}\sum_{j=0}^{S-1}\sum_{k=0}^{S-1} \mathbb{1}(Y_r = l)\mathbb{1}(q_t^r = j)\mathbb{1}(q_{t+1}^r = k) \log a_{jk}^{(l)} + \\
&\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{t=1}^{T}\sum_{j=0}^{S-1}\sum_{o=0}^{|\mathcal{O}|-1} \mathbb{1}(Y_r = l)\mathbb{1}(q_t^r = j)\mathbb{1}(o_t^r = o) \log b_j^{(l)}(o)
\end{aligned} \tag{A.2}$$

Onde S denota o número de estados ocultos da HMM e $|\mathcal{O}|$ denota o número de símbolos observáveis distintos.

Para definir o passo de *Expectation* derivamos a distribuição *a posteriori* das variáveis latentes dado o conjunto de variáveis observáveis. Utilizando a independência condicional obtida a partir do PGM, obtemos:

$$\begin{aligned} &P[Y_r = l, q_t^r = j|\mathbf{o}^r, \tilde{Y_r}] = \\ &= P[q_t^r = j|\mathbf{o}^r, Y_r = l]P[Y_r = l|\mathbf{o}^r, \tilde{Y_r}] \\ &= \gamma_{r,t}^{(l)}(j)\tau_r(l) \end{aligned} \tag{A.3}$$

Onde $\gamma_{r,t}^{(l)}(j)$ é calculado através do algoritmo de Forward-Backward [61] e $\tau_r(l)$ é dado por:

$$\tau_r(l) = \frac{\left(\sum_{j=0}^{S-1} \alpha_{r,T}^{(l)}(j)\right) \varepsilon_{\tilde{Y}_r|l}\ \rho_l}{\sum_{u=0}^{1}\left(\sum_{j=0}^{S-1} \alpha_{r,T}^{(u)}(j)\right) \varepsilon_{\tilde{Y}_r|u}\ \rho_u}. \tag{A.4}$$

Combinamos os resultados anteriores para obter a função auxiliar $Q(\theta, \theta_{old})$ otimizada pelo algoritmo de *Expectation-Maximization*:

$$\begin{aligned} Q(\theta, \theta_{old}) = &\sum_{r=1}^{R}\sum_{l=0}^{1} \tau_r(l)\log\rho_l + \\ &\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{m=0}^{1} \tau_r(l)\mathbb{1}(\tilde{Y_r} = m)\log\varepsilon_{m|l} + \\ &\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{k=0}^{S-1} \tau_r(l)\gamma_{r,1}^{(l)}(k)\log\pi_k^{(l)} + \\ &\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{t=1}^{T-1}\sum_{j=0}^{S-1}\sum_{k=0}^{S-1} \tau_r(l)\xi_{r,t}^{(l)}(j,k)\log a_{jk}^{(l)} + \\ &\sum_{r=1}^{R}\sum_{l=0}^{1}\sum_{t=1}^{T}\sum_{j=0}^{S-1}\sum_{o=0}^{|\mathcal{O}|-1} \tau_r(l)\gamma_{r,t}^{(l)}(j)\mathbb{1}(o_t^r = o)\log b_j^{(l)}(o) \end{aligned} \tag{A.5}$$

Para derivar o passo de *Maximization*, otimizamos a função $Q(\theta, \theta_{old})$ em relação a cada parâmetro de interesse. Diferenciando o Lagrangiano de $Q(\theta, \theta_{old})$ obtêm-se:

$$\rho_l = \frac{\bar{\tau}_l}{R} \qquad \varepsilon_{m|l} = \frac{\sum_{r=1}^{R} \tau_r(l)\, \mathbb{1}(\tilde{Y}_r = m)}{\bar{\tau}_l}$$

$$\pi_k^{(l)} = \frac{\sum_{r=1}^{R} \tau_r(l)\, \gamma_{r,1}^{(l)}(k)}{\bar{\tau}_l} \qquad a_{jk}^{(l)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T-1} \tau_r(l)\, \xi_{r,t}^{(l)}(j,k)}{\sum_{r=1}^{R} \sum_{t=1}^{T-1} \tau_r(l)\, \gamma_{r,t}^{(l)}(j)} \tag{A.6}$$

$$b_j^{(l)}(o) = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T} \tau_r(l)\, \gamma_{r,t}^{(l)}(j)\, \mathbb{1}(o_t^r = o)}{\sum_{r=1}^{R} \sum_{t=1}^{T} \tau_r(l)\, \gamma_{r,t}^{(l)}(j)}$$

Onde $\bar{\tau}_l = \sum_{r=1}^{R} \tau_r(l)$ representa o número efetivo de pares RD associados à classe l.