DESCOBRINDO PERFIS DE TRÁFEGO DE USUÁRIOS: UMA ABORDAGEM NÃO SUPERVISIONADA

Ananda Görck Streit

Dissertação de Mestrado 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 Mestre em Engenharia de Sistemas e Computação.

Orientador: Rosa Maria Meri Leão

Rio de Janeiro
Fevereiro de 2019

DESCOBRINDO PERFIS DE TRÁFEGO DE USUÁRIOS:
UMA ABORDAGEM NÃO SUPERVISIONADA

Ananda Görck Streit

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

Examinada por:

Prof. Rosa Maria Meri Leão, Dr.

Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.

Prof. Renata Cruz Teixeira, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2019

Streit, Ananda Görck

Descobrindo perfis de tráfego de usuários:
uma abordagem não supervisionada/Ananda Görck Streit. – Rio de Janeiro: UFRJ/COPPE, 2019.

VIII, 44 p.: il.; 29, 7cm.

Orientador: Rosa Maria Meri Leão

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 39 – 44.

1. Redes de acesso residencial. 2. Perfis de tráfego. 3. Aprendizado de máquina. I. Leão, Rosa Maria Meri. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

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

DESCOBRINDO PERFIS DE TRÁFEGO DE USUÁRIOS: UMA ABORDAGEM NÃO SUPERVISIONADA

Ananda Görck Streit

Fevereiro/2019

Orientador: Rosa Maria Meri Leão

Programa: Engenharia de Sistemas e Computação

As redes domésticas estão cada vez mais complexas. Portanto, é essencial a elaboração de estratégias inovadoras para caracterizar essa nova demanda. Em particular, entender as características do tráfego gerado pelos usuários é de suma importância para o planejamento da rede. Trabalhos anteriores focam principalmente na Inspeção Profunda de Pacotes (DPI) e/ou consideram padrões predeterminados para classificar os fluxos de tráfego e determinar a aplicação sendo utilizada pelos usuários. Neste trabalho utilizam-se técnicas não supervisionadas de aprendizado de máquina com o objetivo de entender o perfil de tráfego dos usuários. Em parceria com um Provedor de Serviço Internet (ISP), foram coletados dados do tráfego de download e upload de mais de 2.000 roteadores domésticos. Em seguida, é aplicada uma técnica de decomposição de tensores (PARAFAC) para extrair fatores relevantes de uso da rede. Mostra-se como os resultados do PARAFAC e de um algoritmo de clusterização hierárquica simplificam a tarefa de agrupamento de séries temporais com padrões de tráfego diário similares. Também se mostra como novos usuários podem ser classificados a partir da árvore de decisão obtida com a clusterização. Para caracterizar o comportamento dos usuários em períodos maiores que um dia, utiliza-se a informação dos clusters e de um Modelo de Markov Oculto (HMM). Resultados do modelo indicam que os usuários tendem a manter um padrão específico ao longo do tempo, facilitando tarefas de planejamento e gerenciamento da rede.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DISCOVERING USER TRAFFIC PROFILES:
AN UNSUPERVISED APPROACH

Ananda Görck Streit

February/2019

Advisor: Rosa Maria Meri Leão

Department: Systems Engineering and Computer Science

The increasing complexity of home networks calls for novel strategies towards efficient network management and workload characterization. In particular, understanding the characteristics of the traffic generated by users is of paramount importance for network planning. Previous work focuses primarily on Deep Packet Inspection (DPI) and/or considers pre-determined patterns to classify traffic flows and detect the application being accessed by users. In this work we use unsupervised machine learning techniques with the objective of discovering users' traffic profiles. In partnership with an Internet Service Provider (ISP) we collected the download and upload traffic of more than 2,000 home routers of the ISP clients. We then use a tensor decomposition technique (PARAFAC) to extract relevant features from our network traces. We show how the results of PARAFAC and a hierarchical clustering algorithm simplify the task of grouping time series with similar daily traffic patterns. We also show how new users can be classified from the decision tree obtained with clustering. To characterize users' behavior over periods longer than a day, we use the information of the clusters and a Hidden Markov Model (HMM). The results indicate that users tend to maintain a specific pattern over time, facilitating network planning and management tasks.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A crescente complexidade da Internet, expressiva após uma explosão sem precedentes do número de dispositivos IoT conectados a roteadores domésticos, exige novas estratégias para o gerenciamento eficiente da rede. Pesquisadores e profissionais de Provedores de Serviço Internet (ISPs) estão usando técnicas de aprendizado de máquina para entender melhor o comportamento do usuário doméstico. Tradicionalmente esse tema esteve fora do escopo dos ISPs, seja devido a preocupações com a privacidade ou devido à incapacidade de processamento de grandes volumes de dados. No entanto, o comportamento do usuário doméstico é fundamental para lidar com problemas de segurança e de desempenho da rede.

Entender as características do tráfego gerado pelos usuários é de suma importância para uma variedade de aplicações, como detecção de tráfego anômalo, previsão do tráfego futuro e alocação adequada de recursos da rede [1]. Em particular, devido à sua importância para o planejamento da rede, a classificação de tráfego tem sido um assunto popular há muitos anos (e.g, [2–6]).

Trabalhos anteriores empregam técnicas de aprendizado de máquina para analisar a grande quantidade de dados coletada por ferramentas de monitoramento [1]. Porém, eles focam principalmente na Inspeção Profunda de Pacotes (DPI) e/ou consideram padrões predeterminados para classificar fluxos de tráfego em aplicações específicas [7]. A literatura sobre técnicas de aprendizado de máquina não supervisionadas para dados coletados sem DPI ainda é escassa, talvez devido às dificuldades para acessar medições reais de usuários residenciais [8].

Considere, por exemplo, o recente trabalho de Morichetta e Mellia [3], onde o tráfego HTTP é monitorado para extrair URLs e, em seguida, caracterizar o tráfego do usuário. Embora este trabalho compartilhe algumas metas comuns, por razões de privacidade, o conjunto de dados considerado neste trabalho não depende de identificadores de solicitação de objeto, o que naturalmente leva a diferentes tipos de técnicas de aprendizado de máquina para extrair características relevantes dos dados.

Neste trabalho utiliza-se um conjunto de dados coletado em parceria com um Provedor de Serviço de Internet (ISP) localizado no Brasil. Foram reunidos dados do tráfego de download e upload de mais de 2.000 roteadores domésticos, medidos a cada minuto durante 28 dias (de 20 de agosto a 16 de setembro de 2018). Apesar da coleta estar limitada a um único país, acredita-se que este estudo seja relevante para que os ISPs possam estrategicamente melhorar a alocação de banda.

Esta não é a primeira vez que uma coleta de dados desse tipo é realizada com tal granularidade na borda da rede. No trabalho recente de Trevisan *et al.* [7] foram analisados dados de tráfego similares, reunidos por um período de cinco anos em um ISP na Itália. Porém, diferente do estudo realizado aqui, Trevisan *et al.* [7] faz uma investigação macro do acesso a rede; um dos seus objetivos é examinar as mudanças que ocorreram no tráfego agregado dos usuários ao longo desses cinco anos.

Por outro lado, este trabalho realiza uma análise micro do acesso à rede doméstica, respondendo as seguintes questões: (1) Como extrair com eficiência características relevantes do conjunto de dados de tráfego, sem pré-rotular esses dados e preservando a privacidade dos usuários (e.g, assumindo que o tráfego é totalmente criptografado)? (2) Como interpretar e avaliar os resultados obtidos com as ferramentas de aprendizado de máquina?

Embora a caracterização e a classificação do tráfego tenham sido estudadas por muitos anos, recentemente houve uma mudança drástica no tráfego dos clientes trazendo novos desafios aos ISPs [7]. Em primeiro lugar, o desenvolvimento contínuo de novas aplicações e o surto frequente de ataque distintos requerem a aprendizagem de padrões que não foram considerados em esforços anteriores. A aplicação de estratégias de aprendizagem não supervisionadas, por exemplo, é uma vantagem nesse tipo de cenário.

Em segundo lugar, é necessário contornar preocupações com a privacidade, usando apenas dados sobre métricas agregadas de tráfego, tais como taxas de download e upload, atrasos e perdas na rede. Terceiro, para lidar com grandes conjuntos de dados, utilizam-se algoritmos eficientes e interpretáveis, como a decomposição do tensor pelo método PARAFAC seguida de interpretação e avaliação dos grupos de usuários detectados.

Em resumo, as principais contribuições deste trabalho são:

- *Framework para análise do comportamento do usuário ao acessar sua rede doméstica:* Aqui é proposta uma metodologia simples e direta para detectar estruturas temporais e padrões comportamentais da atividade de tráfego de usuários residenciais. Em suma, são extraídas características das séries temporais de download e upload que sugerem padrões diários comuns de tráfego. Como consequência, é possível identificar perfis de comportamento dos usuários em períodos maiores que um dia.

- *Modelo do perfil diário de usuários residenciais:* Utiliza-se uma técnica de decomposição de tensores (PARAFAC) para obter uma representação mais simples das amostras de tráfego diário. O PARAFAC é um método bem estabelecido e tem sido aplicado em áreas como psicometria [9], quimiometria [10, 11] e processamento de sinais, classificação e aprendizado [12, 13]. Uma vantagem do PARAFAC, em comparação com outros métodos de decomposição fatorial (e.g, PCA, SVD ou Tucker), é a garantia de solução única sob condições moderadas. Dessa forma, é possível obter fatores interpretáveis e intrínsicos ao conjunto de dados de tráfego, sem a necessidade de aplicar métodos externos de rotação (e.g, *Varimax* [14]) para determinar o espaço fatorial mais adequado ao problema. O resultado obtido pelo PARAFAC pode ser usado para vários fins: classificação, previsão e clusterização. Este trabalho concentra-se no último e mostra como o PARAFAC simplifica a tarefa de agrupamento de séries temporais em perfis diários.

- *Modelo do perfil de comportamento de usuários residenciais em períodos maiores que um dia:* Elabora-se um Modelo de Markov Oculto (HMM) às sequências de perfis diários detectados a partir do modelo PARAFAC. O modelo final obtido indica que os usuários tendem a manter um padrão específico ao longo do tempo. Por ser um modelo generativo, os padrões identificados podem ser reproduzidos para simulações e testes com base em dados reais de tráfego. Além disso, esse resultado pode facilitar tarefas de planejamento e gerenciamento da rede.

Além desta Introdução, este texto está organizado em mais cinco capítulos. O Capítulo 2 realiza uma revisão bibliográfica sobre o tema. Tem como objetivo fundamentar o entendimento do assunto em estudo e permitir um conhecimento sobre o estado da arte na caracterização de tráfego da Internet.

O Capítulo 3 faz a apresentação de alguns conceitos que servem de base para o entendimento deste texto como um todo. Nele descreve-se o funcionamento do método PARAFAC para decomposição de tensores, e detalha-se o processo de clusterização utilizado para agrupar amostras similares. Também apresenta-se a definição da árvore de decisão aplicada para classificação e previsão de dados, e introduz-se o Modelo de Markov Oculto utilizado para modelar as sequências observadas com estados ocultos discretos.

O Capítulo 4, por sua vez, apresenta o *framework* proposto para a análise do comportamento do usuário ao acessar sua rede doméstica. Os resultados obtidos através da aplicação da metodologia proposta e os modelos desenvolvidos para o descobrimento e a caracterização de perfis de tráfego são detalhados no Capítulo 5.

Por fim, o Capítulo 6 apresenta uma breve síntese do que é tratado ao longo

do texto. Nesse capítulo, mencionam-se também as sugestões e os direcionamentos possíveis para trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

À medida que a quantidade de dispositivos e aplicações acessando a Internet continua a crescer, torna-se cada vez mais importante entender o comportamento do tráfego na rede. A caracterização do tráfego fornece informações para várias atividades de gerenciamento, como planejamento e provisionamento de capacidade, engenharia de tráfego, diagnóstico de falhas, detecção de anomalias e determinação de tarifas [15].

Em vista disso, este capítulo traz uma visão geral sobre os principais trabalhos desenvolvidos recentemente para caracterizar o tráfego da Internet, a fim de permitir ao leitor um entendimento do estado da arte nessa área de pesquisa. A exposição dos trabalhos é realizada em duas seções independentes de acordo com o propósito: a Seção 2.1 apresenta trabalhos que empregam técnicas de decomposição de tensores em dados de tráfego; e a Seção 2.2 discorre sobre os estudos que buscam identificar perfis de tráfego com padrões de comunicação similares.

A Figura 2.1 apresenta um organograma contendo os trabalhos mencionados neste capítulo com base nos seus objetivos e propostas principais.

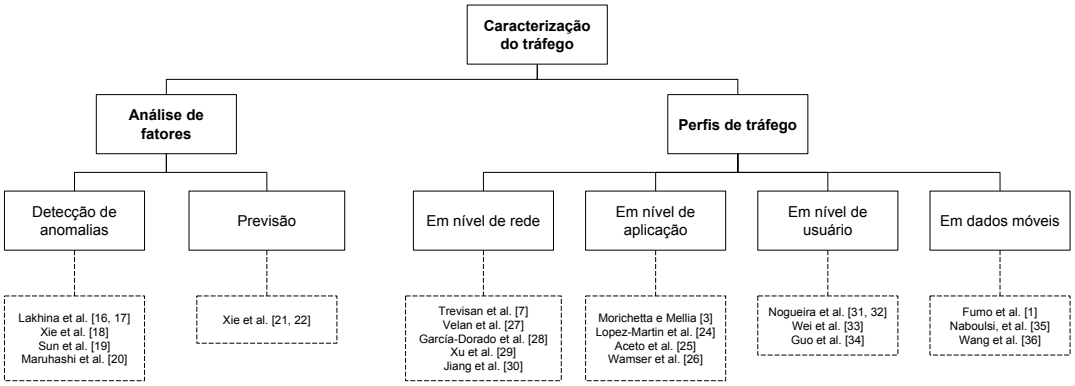

Figura 2.1: Organograma dos principais trabalhos de caracterização do tráfego

2.1 Análise de fatores aplicada em dados de tráfego

O objetivo desta seção é ressaltar que, embora existam diversos trabalhos na literatura que empregam técnicas de decomposição de tensores em dados de tráfego, tanto os objetivos como a forma de aplicação são distintos do que é realizado nesta dissertação. Isso revela a autenticidade deste trabalho, bem como demonstra os potenciais caminhos a serem investigados no futuro.

Alguns trabalhos utilizam métodos de análise de fatores para determinar o comportamento normal do tráfego e, a partir disso, detectar comportamentos anômalos e atividades maliciosas. Por exemplo, Lakhina *et al.* [16, 17] utilizam *Principal Component Analysis* (PCA) para detectar, identificar e quantificar anomalias de tráfego a partir da quantidade de bytes transmitidos em um link [16] ou a partir do volume de tráfego em fluxos de origem-destino [17].

Nessa mesma linha de pesquisa, outros estudos costumam estruturar os dados em tensores de três dimensões (e.g, (nó de origem $\times$ nó de destino $\times$ tempo) ou (nó de origem $\times$ nó de destino $\times$ número de porta)) [18–20]. Os tensores são utilizados para representar objetos de múltiplas dimensões. O objetivo é determinar os padrões normais de tráfego trocado entre pares de nós em diferentes instantes de tempo ou em diferentes portas e encontrar anomalias/eventos ocultos na estrutura multidimensional. Nesse contexto, o recente trabalho de Xie *et al.* [18] propõe um modelo de decomposição de tensores que incorpora grafos para representar informações não lineares e melhorar a acurácia na detecção de anomalias.

Em outro campo de pesquisa, a técnica de decomposição de tensores é empregada para o preenchimento de espaços em branco em séries de tráfego (i.e, previsão de dados) [21, 22]. Os tensores são geralmente incompletos para esse tipo de dado devido à inevitável perda de pacotes durante o monitoramento da rede. O objetivo do preenchimento é viabilizar a aplicação de métodos de aprendizagem que demandem dados completos ou que são extremamente sensíveis à falta de informações.

No trabalho de Xie *et al.* [21] sugere-se a adaptação das técnicas de aproximação de baixo posto habitualmente aplicadas às matrizes para o preenchimento sequencial de tensores. Eles buscam desenvolver uma técnica com baixo custo computacional e também explorar as múltiplas correlações entre as dimensões dos tensores, o que proporciona precisão da informação e consequentemente ajuda a preservar as características latentes presente nos dados de tráfego. Uma adaptação desse método é sugerida por Xie *et al.* [22] para medições dinâmicas, onde os intervalos de coleta podem variar em função das condições da rede.

2.2 Perfis de tráfego

Embora exista um extenso conjunto de trabalhos que trata sobre a caracterização de tráfego em *backbones* - especialmente em termos de propriedades estatísticas (e.g, autossimilaridade [23]), trabalhos mais recentes buscam construir perfis em termos de comportamentos, ou seja, padrões de comunicação em nível de aplicação, em nível de rede e em nível de *hosts*/usuários finais.

Em nível de aplicação, o objetivo principal é identificar serviços pelo padrão de tráfego que eles geram. Com a crescente adoção de protocolos criptografados, *Network Address Translation* (NAT) e portas dinâmicas, abordagens tradicionais para a classificação dos fluxos de tráfego, tais como *Deep Packet Inspection* (DPI) e métodos baseados em porta, estão sendo substituídos por métodos de aprendizado de máquina. Alguns exemplos de classificação de tráfego em nível de aplicação são apresentados a seguir.

Lopez-Martin *et al.* [24] propõem um classificador formado pela combinação de dois métodos de aprendizagem profunda: uma Rede Neural Convolucional (CNN) e uma Rede Neural Recorrente (RNN). Ele apresenta resultados altos de detecção (acima de 95%) e funciona com um número pequeno de atributos extraídos dos pacotes. No trabalho de Aceto *et al.* [25] também são testadas diferentes técnicas de aprendizagem profunda, porém a detecção é realizada no tráfego gerado em dispositivos móveis.

Outra abordagem é proposta por Wamser *et al.* [26]. Eles apenas utilizam dados de redes residenciais e classificam os fluxos com base no comportamento dos usuários ao acessar a rede. O comportamento dos usuários é registrado de acordo com o uso de portas e endereços IP para facilitar a distinção entre tráfego *Peer-to-Peer* (P2P), Web e *streaming*. Eles verificam, por exemplo, que 1,6 Mbps é constantemente usado pelo tráfego P2P durante o dia. Os tráfegos Web e *streaming*, por outro lado, costumam variar ao longo do dia porque precisam de interação do usuário.

Já o recente trabalho de Morichetta e Mellia [3] aplica um método não supervisionado para a identificação de serviços. Eles monitoram o tráfego HTTP e extraem URLs para detectar substrings comuns no caminho do objeto, mesmo quando os nomes de domínio são distintos. Eles argumentam que padrões e expressões regulares tipicamente ocorrem em serviços que utilizam a mesma plataforma Web. Os resultados mostram que a proposta também permite descobrir tráfego *malware* gerado por máquinas infectadas.

Em sua maioria, os trabalhos em nível de rede focam na caracterização do tráfego agregado gerado por um conjunto de usuários compartilhando a mesma rede. Por exemplo, o trabalho de Velan *et al.* [27] analisa os padrões de comunicação de redes individuais da Universidade Masaryk, situadas em diferentes partes do campus. Eles

mostram que é possível distinguir as redes a partir de características simples como os padrões dia-noite e dia da semana no tráfego.

García-Dorado *et al.* [28] comparam clientes de ISPs e tecnologias de acesso diferentes a partir de *traces* coletados em um período de 21 meses. Eles verificam que os padrões diários de tráfego são regulares dentro de um mesmo país ou em função de uma tecnologia específica. Assim, também concluem que a caracterização do tráfego pode se alterar significativamente entre redes diferentes.

Trevisan *et al.* [7] realiza um estudo em dados coletados durante cinco anos em um ISP da Itália. Os pesquisadores verificaram que o tráfego de download diário dos usuários mais que duplicou durante o período de análise. Além disso, o tráfego P2P reduziu significativamente e foi trocado por conteúdos de vídeo acessíveis e dentro da legalidade.

O trabalho de Xu *et al.* [29] foca nos padrões de comportamento de grupos de *hosts* finais nos mesmos prefixos de rede. O objetivo é concentrar-se no comportamento do tráfego gerado por grupos de usuários próximos e consequentemente reduzir o número de perfis para análise em comparação com um estudo em nível de usuário. Eles utilizam gráficos bipartidos para representar os padrões de comunicação entre os *hosts* de origem e de destino e realizam clusterização espectral nas matrizes de similaridade obtidas pela projeção desses gráficos. Eles mostram que a maioria dos usuários permanecem no mesmo *cluster* ao longo do tempo. Além disso, os experimentos sugerem que a análise do tráfego de *clusters* em um mesmo prefixo facilita a detecção de tráfego anômalo.

A pesquisa desta dissertação também realiza clusterização para separar os usuários em grupos com comportamentos similares de tráfego (veja o Capítulo 4). A principal diferença com o trabalho de Xu *et al.* [29], descrito acima, é que a análise é realizada em nível de usuário. Assim, os *clusters* não estão restritos à localização dos *hosts* na rede. Além disso, os grupos são determinados em função do volume de tráfego gerado ao longo do dia pelos usuários e não com base nos endereços IP de origem e de destino dos pacotes trocados entre os *hosts*.

Ainda tratando sobre trabalhos em nível de rede, Jiang *et al.* [30] aplicam o algoritmo de clusterização K-Means em séries temporais representando o volume de tráfego agregado produzido a cada hora em diferentes prefixos. Os resultados mostram perfis de tráfego diário com comportamentos diferentes ao longo do dia, similar a um dos objetivos desta dissertação, porém aplicado em nível de rede. Eles também estudam outras característica do tráfego de download agregado, incluindo o volume diário total de tráfego, a distribuição de tráfego em função do tipo de aplicação, etc.

Por fim, a seguir são descritos alguns trabalhos que tratam sobre perfis de comportamento em nível de usuário. Note que a granularidade é a principal diferença

entre as abordagens em nível de rede e em nível de usuário. Conforme mencionado por Nogueira *et al.* [31, 32], é vantajoso para os ISPs terem um conhecimento detalhado dos principais tipos de comportamento de usuários individuais, que podem permanecer ocultos se o tráfego for analisado em um nível agregado.

Por exemplo, com esse tipo de conhecimento um ISP pode otimizar a sua rede mesclando no mesmo nó da rede os usuários cujos períodos de maior utilização do tráfego estão em intervalos de tempo separados. Além disso, o conhecimento sobre o comportamento dos usuários auxilia na definição de políticas tarifárias baseadas em horários. Compartilhando desses objetivos, esta dissertação também é voltada para o aprendizado de perfis em nível de usuário.

O trabalho de Nogueira *et al.* [32] aplica clusterização hierarquica aglomerativa em séries de tráfego geradas por diferentes usuários. Eles separam as séries em três grandes grupos. O primeiro *cluster* contém usuários com altas taxas de transferência em todos os períodos do dia; os usuários do segundo *cluster* têm baixas taxas de transferência pela manhã e altas taxas de transferência no período da tarde; e o terceiro *cluster* contém usuários com baixas taxas de transferência em todos os períodos do dia.

Apesar dos objetivos comuns aos desta dissertação, o foco de Nogueira *et al.* [32] é a classificação dos usuários nos três perfis (*clusters*) encontrados. Eles comparam o uso de Análise Discriminante e Redes Neurais para realizar essa tarefa. Porém, esta dissertação não fica restrita a análise do comportamento diário dos usuários e apenas emprega a classificação como um método complementar para a inclusão de novas séries na análise.

Wei *et al.* [33] aplicam um algoritmo aglomerativo para agrupar *hosts* finais em *clusters* com o objetivo de detectar anomalias. Eles utilizam vários atributos de tráfego, incluindo o volume total de bytes diário e conexões TCP e UDP observadas em uma janela de tempo. Guo *et al.* [34] propõem um algoritmo de clusterização hierarquica aglomerativa com menor complexidade de tempo que o algoritmo clássico. Eles agrupam os *traces* a fim de descobrir o padrão de preferências dos usuários e recomendar os serviços mais adequados. Também analisam as preferências dos usuários em função da duração do acesso ao serviço e do tráfego total requisitado.

Vale ressaltar também as pesquisas voltadas para a caracterização do comportamento de tráfego em redes móveis [1, 35, 36]. Os trabalhos nessa área normalmente buscam analisar e categorizar em perfis o comportamento dos usuários acessando a rede móvel, avaliando em conjunto os aspectos temporais de acesso com os padrões de mobilidade em uma área geográfica.

Capítulo 3

Conceitos básicos

Este capítulo faz a apresentação de alguns conceitos que servem de base para o entendimento deste texto. A Seção 3.1 introduz o método PARAFAC, empregado para a obtenção de fatores que descrevam os dados originais de forma mais compacta. A Seção 3.2 detalha o processo de clusterização utilizado para agrupar amostras similares. A Seção 3.3 apresenta a definição da árvore de decisão, aplicada para classificação e previsão de dados. Por último, a Seção 3.4 faz uma introdução ao Modelo de Markov Oculto (HMM), conforme originalmente proposto por Rabiner [37].

3.1 Análise de fatores

Um dos principais objetivos da análise fatorial é decompor um *array* em um conjunto de fatores (variáveis latentes) e cargas (*loadings*) que descrevam os dados de forma mais compacta em comparação com o conjunto original. A sua aplicação é principalmente vista como uma ferramenta exploratória para detecção de estruturas (fatores) subjacentes interpretáveis que expliquem as correlações entre o conjunto de variáveis originais.

Dessa forma, o conceito-chave da análise fatorial é que múltiplas variáveis observadas têm padrões semelhantes de respostas, estando todas associadas a um fator que não é diretamente mensurado. Os *loadings* expressam a relação entre cada variável e cada um dos fatores subjacentes detectados.

O termo *análise de fatores* foi introduzido por Thurstone em 1931 [38]. Entre as diversas abordagens de decomposição de dados, o *Principal Component Analysis* (PCA) [39] e o *Singular Value Decomposition* (SVD) [40] estão entre os modelos bilineares mais populares. Já os métodos multilineares mais conhecidos são o *Parallel Factor Analysis* (PARAFAC) [41] e o Tucker3 [9].

Neste trabalho, utiliza-se o PARAFAC por sua simplicidade e garantia de solução única sob condições moderadas [42, 43]. Ele é apresentado na Subseção 3.1.1. Na Subseção 3.1.2 realiza-se uma breve comparação entre o PARAFAC e outros métodos

populares de extração de fatores com o objetivo de motivar e justificar a escolha do PARAFAC.

3.1.1 O método PARAFAC

O PARAFAC, por ser um método multilinear, decompõe tensores para obter fatores subjacentes que representem o conjunto de dados originais. Tensores são usados para representar *arrays* N-dimensionais, servindo como generalizações de escalares (que não têm índices), vetores (que têm exatamente um índice) e matrizes (que têm exatamente dois índices) para um número arbitrário de N índices. Portanto, sob esta abordagem, um vetor é um tensor unidimensional ou de primeira ordem e uma matriz é um tensor bidimensional ou de segunda ordem.

Um tensor de ordem 3 possui a forma de um cubo. Nesse caso, é possível estruturar os dados, por exemplo, em fatias de matrizes que representem individuos e suas características (como altura, idade, peso, etc.) em diferentes instantes de tempo. A Figura 3.1 exemplifica um tensor desse tipo. A notação tensorial é muito parecida com a notação matricial, com uma letra maiúscula representando o tensor e letras minúsculas com números inteiros subscritos representando valores escalares dentro do tensor.

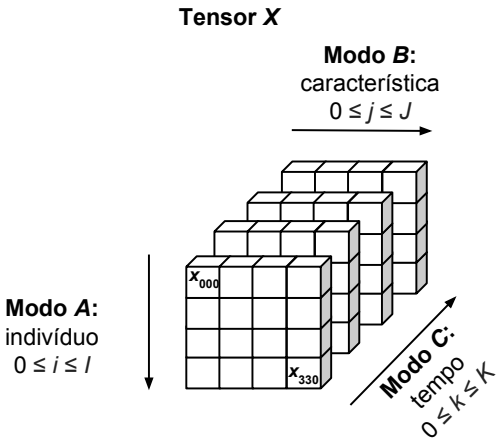

Figura 3.1: Exemplo de um tensor de ordem 3

A seguir, a discussão é limitada a tensores de ordem 3 para simplificar a apresentação, porém a maioria dos resultados também é válida para tensores de qualquer ordem. Seja $X \in \mathbb{R}^{I \times J \times K}$ um tensor de ordem 3, tendo x_{ijk} como um dos seus elementos. A decomposição PARAFAC é dada como,

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} + e_{ijk}, \tag{3.1.1}$$

onde R é o número de fatores; e, a_{ir}, b_{jr} e c_{kr} são as cargas (ou *loadings*) do fator r correspondentes aos modos A, B e C, respectivamente. Os termos *cargas* e *loadings* são usados como sinônimos ao longo deste trabalho. Os residuais são denotados por e_{ijk}. Esta equação é mostrada graficamente na Figura 3.2 para quatro fatores ($R = 4$).

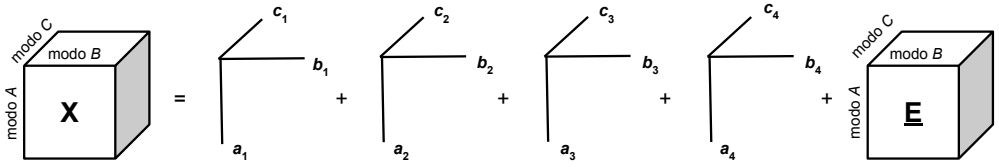

Figura 3.2: Representação gráfica de um modelo PARAFAC com quatro fatores

O objetivo do método é calcular as cargas que minimizem a soma dos quadrados dos resíduos utilizando o algoritmo de mínimos quadrados alternantes (*Alternanting Least Squares* (ALS)) [44]. No algoritmo ALS as matrizes de cargas de cada um dos fatores são estimadas uma de cada vez, mantendo as outras matrizes fixas. Esse processo é repetido até que um critério de convergência seja satisfeito ou até que não ocorra mudança nas estimativas.

Existem alguns passos necessários para implementação do método PARAFAC; veja [11] e [45] para algumas recomendações. Para determinar o número adequado de fatores (R) a serem extraídos de um dado conjunto de dados, pode-se analisar o percentual de variância explicada do modelo. Esse valor é calculado a partir da soma dos quadrados dos resíduos (E) em relação à soma dos quadrados dos dados (X).

O número de fatores também pode ser definido complementarmente usando outros métodos. O trabalho de Stedmon *et al.* [11] sugere a aplicação do método *Split-Half Validation* (SV) [45] junto com o *Tucker Congruence Coefficient* (TCC) [46] para determinar o número de fatores (R) e, ao mesmo tempo, avaliar se a solução é única e generalizável para outro conjunto de dados similar. O método SV consiste em dividir as amostras relacionadas a um dos modos (a explicação a seguir é restrita ao modo A) de forma aleatória em quatro grupos (G_1, G_2, G_3 e G_4) e testar a similaridade entre os modelos PARAFAC dos subconjuntos independentes ($G_1 + G_2$) vs. ($G_3 + G_4$) e ($G_1 + G_3$) vs. ($G_2 + G_4$).

Para cada uma das validações de similaridade têm-se dois modelos, m_1 e m_2. A diferença entre eles é medida pelo TCC ($\phi_b(r)$ e $\phi_c(r)$, eq. 3.1.2) usando o vetor de cargas relacionado aos modos B ($\boldsymbol{b_r}$) e C ($\boldsymbol{c_r}$),

$$\phi_b(r) = \frac{\sum_{j=1}^{J} b_{jr}^{m_1}\ b_{jr}^{m_2}}{\sqrt{\sum_{j=1}^{J}(b_{jr}^{m_1})^2 \sum_{j=1}^{J}(b_{jr}^{m_2})^2}}, \quad \phi_c(r) = \frac{\sum_{k=1}^{K} c_{kr}^{m_1}\ c_{kr}^{m_2}}{\sqrt{\sum_{k=1}^{K}(c_{kr}^{m_1})^2 \sum_{k=1}^{K}(c_{kr}^{m_2})^2}}, \quad (3.1.2)$$

respectivamente, onde $1 \leq r \leq R$. Dessa forma, obtém-se um ϕ para cada fator e para cada um dos dois modos. Como a comparação é realizada entre modelos gerados a partir de subconjuntos diferentes do modo A, não se calcula ϕ para este modo ($\boldsymbol{a_r}$). O objetivo é saber se os fatores latentes são similares em $\boldsymbol{b_r}$ e $\boldsymbol{c_r}$ mesmo com populações diferentes em A. Quanto mais próximo ϕ é de um, mais semelhantes são as cargas dos fatores. Os resultados de Lorenzo-Seva e Ten Berge [46] sugerem um valor mínimo de $0,95$ para que os dois vetores possam ser considerados iguais.

Caso nenhuma das duas comparações apresente o mesmo padrão de carga (baixa congruência), conclui-se que o conjunto de dados é inadequado para esse tipo de análise (i.e, não apresenta solução única) ou que o número de fatores deve ser reduzido para deixar de representar informação ruidosa e inerente a cada um dos subconjuntos modelados.

3.1.2 PARAFAC e outros métodos de decomposição tensorial

O método PARAFAC, adotado neste trabalho, é um dos vários métodos de decomposição tensorial já propostos na literatura. Os três principais concorrentes são o Tucker3, o PCA e o SVD. Tanto o PARAFAC como o Tucker3 podem ser aplicados diretamente em um tensor com um número qualquer de dimensões. A única limitação é a complexidade computacional requerida em decomposições de tensores de ordens superiores.

Os métodos de análise bilineares, como o PCA e o SVD, estão restritos a tensores de segunda ordem (matrizes). Dessa forma, caso os dados estejam armazenados em tensores de ordem superior, pode-se simplesmente reordenar os elementos em uma matriz, processo conhecido como matricização, para a compreensão da estrutura dos dados.

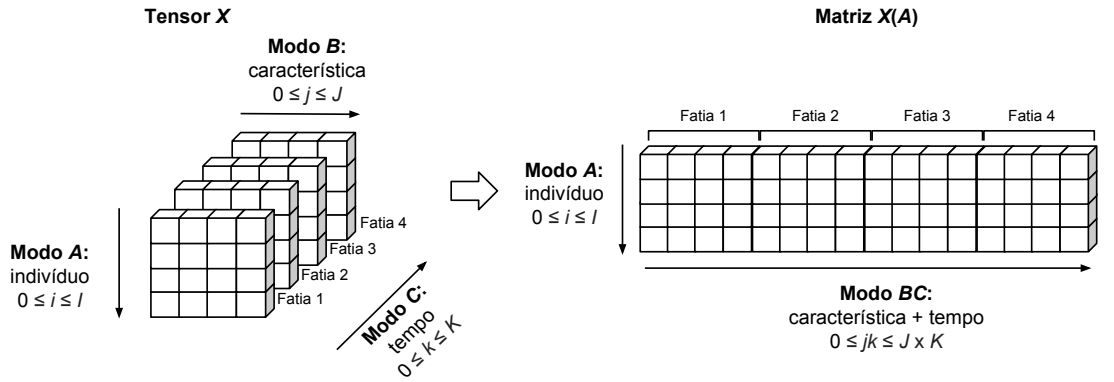

Figura 3.3: Matricização no modo A do tensor X

Usando como exemplo o tensor de ordem 3 representado na Figura 3.1; ilustra-se, na Figura 3.3, o processo de matricização no primeiro modo do tensor X (modo A).

Nesse exemplo, o tensor $X \in \mathbb{R}^{I \times J \times K}$ é desdobrado e uma matriz de tamanho $I \times JK$ é formada, denotada por $X(A)$.

Porém, em diversas aplicações é natural e benéfico armazenar e analisar os dados com natureza multidimensional com métodos multilineares como o PARAFAC e o Tucker3. Isso porque a necessidade de transformar os dados, que naturalmente podem ser estruturados como tensores, em matrizes, faz com que as variáveis nos modos desdobrados (Modos B e C no exemplo da Figura 3.3) se misturem e passem a estar associadas não apenas a um, mas a muitos elementos de um vetor de cargas (*loadings*) [44].

Retomando o exemplo da Figura 3.3. Conforme visto na Subseção 3.1.1, ao aplicar-se o PARAFAC no tensor de ordem 3 (X) obtêm-se os vetores de cargas $\boldsymbol{a_r}$, $\boldsymbol{b_r}$ e $\boldsymbol{c_r}$, um para cada modo e para cada fator r. Considere agora que a variável t_0 indica a Fatia 1 do modo C. Na decomposição pelo PARAFAC, t_0 apenas possui um valor de carga para cada um dos fatores considerados. Por outro lado, ao aplicar-se um método bilinear qualquer na matriz $X(A)$, obtêm-se os vetores de cargas dados por $\boldsymbol{a_r}$ e $\boldsymbol{bc_r}$. Veja que a variável t_0, nesse caso, está associada a mais de um valor de carga para cada fator.

A interpretabilidade dos resultados também é afetada quando aplica-se um método bilinear na matriz $X(A)$. Um padrão diferente para cada uma das fatias do modo C é obtido, mesmo quando os fatores subjacentes possuem uma estrutura similar em todas as fatias do tensor X. Um exemplo real é apresentado no Capítulo 5. Em razão disso, Bro [44] ressalta que modelos multilineares são menos complexos do que modelos bilineares nesse tipo de contexto, mesmo que sejam conceitualmente mais difíceis de compreendê-los.

Por fim, a principal propriedade do PARAFAC em comparação com outros métodos de decomposição tensorial é a sua capacidade de determinar fatores únicos e intrínsicos ao conjunto de dados, sem a necessidade de utilizar nenhum outro método externo de rotação (e.g, *Varimax* [14]) para determinar o espaço fatorial mais adequado ao problema. Isso é especialmente vantajoso dado que diferentes técnicas de rotação muitas vezes dão origem a diferentes hipóteses científicas sobre estruturas subjacentes em uma determinada aplicação [47].

Conforme mencionado por Bro [44], o significado matemático da unicidade é que um modelo PARAFAC não pode ser rotacionado sem decrescer o percentual de variância explicada do modelo. Pelo contrário, ao se rotacionar o espaço fatorial de modelos bilineares, o percentual de variância explicada do modelo não é alterado.

O Tucker3, visto como uma generalização do SVD para tensores de ordem superior [48], também permite a rotação de fatores sem alterar o valor da variância explicada. Isso porque ele é um modelo mais complexo e menos restritivo que o PARAFAC [42]. Para uma discussão mais detalhada sobre as diferenças entre os

modelos de decomposição tensorial mencionados, veja [42, 44]. Para provas matemáticas da propriedade de unicidade do PARAFAC, veja [49].

3.2 Clusterização hierárquica aglomerativa

O problema de clusterização (agrupamento) consiste em agrupar as amostras de um conjunto de dados de modo que as amostras mais similares permaneçam no mesmo grupo (*cluster*) e as amostras menos similares estejam em grupos distintos. Há vários métodos para se especificar o critério de similaridade e o procedimento apropriado no problema de clusterização. A escolha depende dos objetivos do agrupamento e do domínio da aplicação.

Neste trabalho, optou-se pela clusterização hierárquica aglomerativa [50], umas das técnicas mais populares para realizar esse tipo de agrupamento. Esse método utiliza uma abordagem *bottom-up*, começando com cada uma das amostras em um *cluster* próprio e sucessivamente agrupando-as a medida que se sobe na hierarquia. O algoritmo termina quando todas as amostras pertencem a um único *cluster*. No final, esse processo de agrupamento gera uma árvore conhecida como dendrograma.

O dendograma tem como principal vantagem permitir a visualização dos grupos gerados em cada um dos diferentes níveis da árvore. Assim, em comparação com outros métodos de clusterização como, por exemplo, o K-Means, não é necessário determinar o número de *clusters* desejado antes de se executar o algoritmo.

Há vários métodos para se analisar a similaridade entre as amostras no problema de clusterização. Um dos mais utilizados é o método de variação mínima de Ward [51], empregado para selecionar o melhor par de *clusters* a serem mesclados em cada etapa do algoritmo de clusterização hierárquica. O par escolhido busca minimizar a soma das distâncias quadradas entre as amostras e o centróide do *cluster*. A distância ao quadrado é definida como [52]:

$$E_K^2 = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \sum_{r=1}^{R} \left(a_{irk} - \bar{a}_{rk}\right)^2 , \tag{3.2.1}$$

onde a_{irk} é o valor associado à variável r da amostra i pertencente ao *cluster* k, n_k é o número de amostras no *cluster* k, e $\bar{a}_{rk} = \sum_{i=1}^{n_k} a_{irk}/n_k$.

3.3 Árvore de decisão

As árvores de decisão [53] são modelos gerados de forma supervisionada a partir de um conjunto de treinamento. Ou seja, elas são construídas a partir de dados com valores de entrada e de saída. A entrada usualmente corresponde a uma matriz re-

lacionando um conjunto de amostras a um conjunto de variáveis. A saída representa a classe relativa a cada uma das amostras.

Além de serem apropriadas para tarefas de classificação, as árvores também estão entre os mais populares algoritmos de inferência. Assim, elas também são úteis para avaliar a influência relativa das variáveis do conjunto de treinamento na tomada de decisão.

A árvore possui três tipos de nós: (1) o nó raiz, representando toda a população de amostras; (2) os nós internos, expressando uma das escolhas possíveis alcançáveis naquele nível; e, (3) os nós folhas, produzindo o resultado final das decisões. Cada nó interno é associado a um critério de divisão e cada nó folha é ligado a uma classe.

O algoritmo de treinamento constrói a árvore de forma recursiva, de cima para baixo, de forma a identificar a variável mais relevante para a classificação das amostras disponíveis em cada um dos nós da árvore. A decisão sobre a divisão dos nós é determinada pelo coeficiente de Gini [53]. Ele pode ser calculado somando-se a probabilidade p_k de uma amostra de classe k ser escolhida vezes a probabilidade $\sum_{q \neq k} p_q = 1 - p_k$ de um erro na sua categorização.

O objetivo é que os nós a serem obtidos pelo processo de divisão sejam minimamente impuros/heterogêneos. Ou seja, o coeficiente de Gini em cada um dos nós resultantes deve alcançar o menor valor possível dentre todas as possibilidades de divisão. Quando apenas existirem amostras de uma única classe em um nó, o coeficiente de Gini atinge seu valor mínimo (zero).

3.4 Modelo de Markov Oculto

Os Modelos de Markov Oculto (HMM) surgiram originalmente em aplicações de reconhecimento de fala [37]. Desde então eles têm sido amplamente utilizados em muitas outras áreas, incluindo processamento de sinais, inteligência artificial, biologia computacional, finanças, processamento de imagens e diagnóstico médico [54].

Um HMM é um modelo probabilístico composto por dois processos estocásticos. Um deles se refere aos resultados observados de um sistema, que, por sua vez, é dependente de um estado latente, não observado, usualmente representado por uma cadeia de Markov. A cadeia de Markov é constituída por um conjunto de estados interconectados $\mathbf{S} = \{S_0, S_1, \ldots, S_N\}$. Tanto a estrutura quanto o número de estados N da cadeia são definidos pelo usuário de acordo com algum conhecimento prévio do sistema sendo estudado.

Os estados ocultos da cadeia satisfazem a propriedade de Markov de primeira ordem. Ou seja, o estado no tempo t, denotado por s_t, depende apenas do estado s_{t-1}, sendo independente de todos os outros estados anteriores $(s_{t-2}, s_{t-3}, \ldots)$. Seja T o tempo total de observação, a sequencia de observaçôes $\mathbf{O} = \{O_0, O_1, \ldots, O_T\}$ tam-

bém satisfaz a propriedade de Markov de primeira ordem com relação aos estados. Isto é, dado um estado S_i, onde $0 \leq i \leq N$; a observação correspondente de S_i é independente de todos os outros estados e observações. Assim, por essa propriedade, entende-se que os estados, em qualquer instante de tempo, encapsulam tudo o que se precisa saber sobre o passado com o objetivo de prever o futuro do processo [55].

Um HMM é completamente determinado por um vetor de probabilidades inicial $\boldsymbol{\pi}$, uma matriz de probabilidade de transição A entre os estados ocultos da cadeia e a matriz de probabilidade de emissão de símbolos B em cada estado oculto S_i. Esses parâmetros são estimados iterativamente utilizando-se o algoritmo Baum-Welch [56].

A tarefa do algoritmo Baum-Welch é ajustar os parâmetros de um modelo HMM, denotado por λ, com o objetivo de maximizar a probabilidade da sequência observada $P(\mathbf{O}|\lambda)$. Para calcular a probabilidade da uma sequência utiliza-se o método *Forward-Backward*. Considere o modelo $\lambda = (\pi, A, B)$, a sequência de observações $\mathbf{O}$ e a sequência de estados ocultos correspondente $s_0, s_1, \ldots, s_T$, onde cada $s_i \in \mathbf{S}$. As probabilidades *Forward* e *Backward* são definidas como

$$\alpha_t(i) = P[O_1, O_2, ..., O_t, s_t = S_i|\lambda] \quad \text{e} \tag{3.4.1}$$

$$\beta_t(i) = P[O_{t+1}, O_{t+2}, ..., O_T|s_t = S_i, \lambda], \text{ respectivamente.} \tag{3.4.2}$$

Ou seja, $\alpha_t(i)$ é a probabilidade da sequência de observações parciais até o tempo t e o estado S_i no tempo t, dado o modelo λ. E $\beta_t(i)$ é a probabilidade da sequência de observações parciais de $t+1$ até o fim do período de observação, dado o estado S_i no tempo t e o modelo λ. Assim, $P(\mathbf{O}|\lambda)$ pode ser calculado por uma das duas probabilidades,

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \quad \text{ou} \quad P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \pi_i \beta_1(i) B(i, O_1). \tag{3.4.3}$$

A partir dessas probabilidades, é possível determinar algumas outras medidas de interesse. Uma delas é a probabilidade de estar no estado S_i no tempo t, dada a sequência de observações $\mathbf{O}$ e o modelo λ, e é denotada por $\gamma_t(i)$. A outra variável, denotada por $\xi_t(i, j)$, refere-se a probabilidade de estar no estado S_i no tempo t e no estado S_j no tempo $t+1$, dada a sequência de observações $\mathbf{O}$ e o modelo λ. Elas são definidas da seguinte forma:

$$\begin{aligned}\gamma_t(i) &= P[s_t = S_i|\mathbf{O}, \lambda] \\ &= \frac{\alpha_t(i).\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i).\beta_t(i)} \end{aligned} \tag{3.4.4}$$

$$\begin{aligned}\xi_t(i,j) &= P[s_t = S_i, s_{t+1} = S_j|\mathbf{O}, \lambda] \\ &= \frac{\alpha_t(i).A(i,j).B(j,O_{t+1}).\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i).A(i,j).B(j,O_{t+1}).\beta_{t+1}(j)} \end{aligned} \tag{3.4.5}$$

A partir de $\gamma_t(i)$ é possível determinar qual é o estado mais provável para qualquer instante de tempo t:

$$s_t = \underset{1 \leq i \leq N}{argmax}[\gamma_t(i)],\ 1 \geq t \geq T \tag{3.4.6}$$

Pode ocorrer da sequência de estados mais prováves obtidos pela Equação 3.4.6 não ser válida e realizável pelo modelo λ. Tanto a estrutura da cadeia de Markov quanto a matriz A do modelo parametrizado podem inviabilizar transições entre estados específicos. Para considerar apenas sequências de estados válidas, pode-se definir o caminho de estados mais provável $s = s_1, s_2, \ldots, s_T$ utilizando-se o algoritmo de Viterbi [57].

Com base nas equações apresentadas, explica-se o funcionamento do algoritmo Baum-Welch a seguir. Inicialmente, define-se de forma arbitrária os parâmetros iniciais π, A e B do modelo λ. Com base nesses valores, é possivel computar as probabilidades $\alpha_t(i)$ e $\beta_t(i)$, e consequentemente as probabilidades $\gamma_t(i)$ e $\xi_t(i,j)$. Em seguida, o algoritmo Baum-Welch reestima os parâmetros do modelo de acordo com as seguintes fórmulas:

$$\overline{\pi}(i) = \gamma_1(i) \tag{3.4.7}$$

$$\overline{A}(i,j) = \frac{\sum_{t=1}^{T-1} \varepsilon_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.4.8}$$

$$\overline{B}(i,k) = \frac{\sum_{t=1, O_t = v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.4.9}$$

Note que o v_k da Equação 3.4.9 corresponde a um dos símbolos possíveis de ser observado pelo modelo do sistema. Os símbolos podem ser tanto discretos como contínuos. No caso discreto, define-se um conjunto possível de símbolos observáveis, dado por $\mathbf{V} = \{v_1, v_2, ..., v_M\}$. No caso contínuo, determina-se uma função de densidade de probabilidade de emissão em cada um dos estados S_i.

Após a atualização, obtém-se um conjunto de novos parâmetros do modelo $\overline{\lambda} = (\overline{\pi}, \overline{A}, \overline{B})$. O processo se repete até que as mudanças nos parâmetros sejam menores que um limite predefinido. Veja que o algoritmo Baum-Welch apenas converge para um ótimo local, o que significa que os parâmetros do modelo não são ótimos no sentido global. Dessa forma, diferentes inicializações podem levar a ótimos locais melhores.

Capítulo 4

Metodologia geral

Uma das principais contribuições deste trabalho é propor um *framework* simples e direto para a análise do comportamento do usuário ao acessar sua rede doméstica. Este capítulo resume os principais passos da metodologia adotada, conforme ilustrado na Figura 4.1.

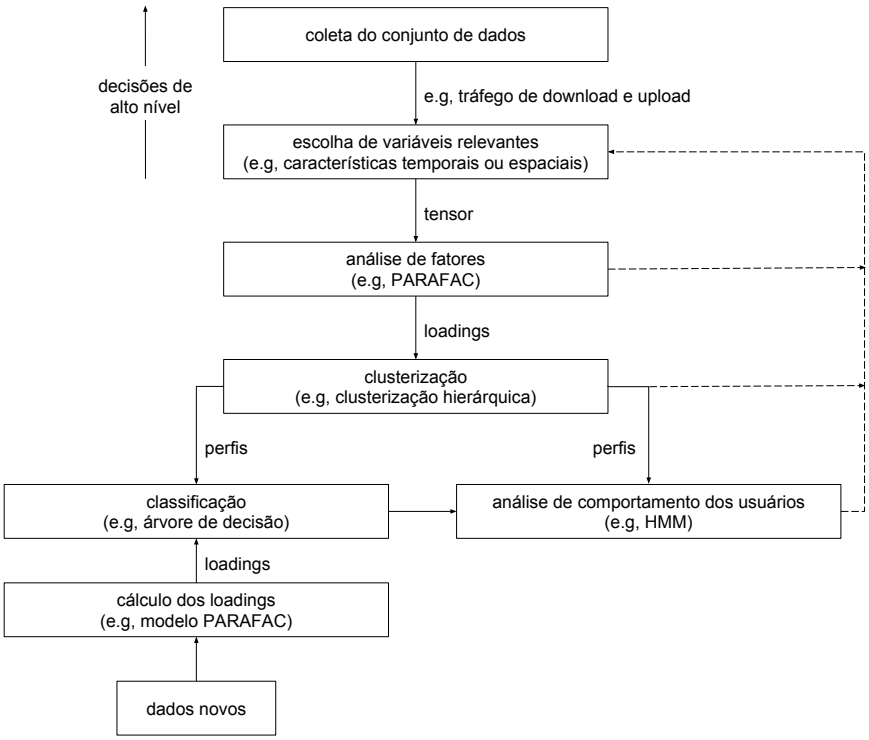

Figura 4.1: *Framework* para aprender perfis de tráfego

Em primeiro lugar deve-se coletar dados dos roteadores residenciais. A Seção 4.1 reune informações sobre como é implementado o ambiente de medição necessário para a coleta das métricas utilizadas na análise do usuário ao acessar a sua rede deméstica. Em seguida, na Seção 4.2, são descritos os passos restantes necessários para a identificação dos perfis de tráfego de usuários domésticos.

Em resumo, no primeiro passo são definidas as variáveis representadas em cada dimensão do tensor a partir do conjunto de dados coletado. Em seguida, utiliza-se o método de decomposição tensorial PARAFAC para extrair características relevantes dos dados, também conhecidas como *loadings*. Na terceira etapa, aplica-se clusterização usando como variáveis os *loadings* obtidos com o método de decomposição tensorial. Posteriormente, é realizada a modelagem do comportamento do usuário usando um Modelo de Markov Oculto (HMM). A metodologia também permite a classificação de novas séries a partir do cálculo dos seus *loadings*, empregando-se um algoritmo de classificação.

4.1 Ambiente de medição e coleta de dados

O conjunto de dados usado na análise experimental consiste no tráfego de download e upload de residências individuais. As amostras de tráfego de download (upload) contém o tráfego total (número de bytes) recebido (enviado) em intervalos fixos de um minuto durante um determinado período de tempo. Neste trabalho, consideram-se amostras de tráfego coletadas durante 28 dias, de 20 de agosto a 16 de setembro de 2018, diretamente de 2.219 roteadores domésticos de diferentes usuários do ISP parceiro.

A coleta foi realizada em roteadores domésticos sem fio que servem como *gateways* domésticos para o provedor de Internet, conforme mostrado na Figura 4.2. O roteadores executam o OpenWrt[58], uma versão do Linux adaptada para sistemas embarcados. Cada roteador possui um *software* de código aberto para coletar e enviar informações para um servidor.

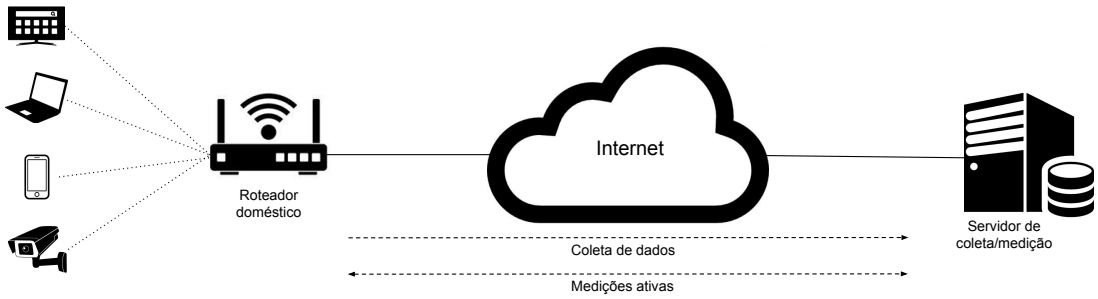

Figura 4.2: Ambiente de medição

Neste estudo não se faz suposições sobre o tipo de dispositivos que formam as redes domésticas individuais. Porém, pelas informações obtidas na coleta de dados, sabe-se que a atividade de tráfego nos roteadores é gerado paralelamente por diversos dispositivos projetados para atender às demandas específicas dos usuários, incluindo computadores pessoais, *smartphones*, videogames, TVs *smart* e outros aparelhos

inteligentes.

Como os roteadores residenciais possuem baixa memória, os resultados das medições são enviados periodicamente ao servidor de coleta. Além da coleta passiva do número de bytes enviado e recebido nos roteadores, também são realizadas medições ativas, como latência e perda. Assim, durante a execução de medições ativas, o servidor não apenas recebe os resultados obtidos como também participa do processo de medição. Uma abordagem similar utilizada para coleta de dados é adotada pela empresa SamKnows [59] e pelo projeto BISmark [60].

4.2 O passo a passo da metodologia

Nesta seção detalham-se os principais passos da metodologia adotada para a obtenção de perfis de uso da rede a partir do conjunto de dados de tráfego coletado.

Passo 1: *escolha de variáveis relevantes*

Uma série temporal de tráfego de download (upload) representa o total de tráfego recebido (enviado) a cada minuto por um determinado usuário em um determinado dia. Denomina-se portanto uma série temporal como um par usuário-dia ou simplesmente um par UD. Define-se I como o número de pares UD do conjunto de dados de tráfego.

Conforme citado no início deste capítulo, utiliza-se a análise fatorial visando extrair as características relevantes dos dados de tráfego dos usuários. Para conhecer o perfil de tráfego dos usuários, é necessário analisar, em conjunto, a quantidade de tráfego de download e upload gerado por cada usuário ao longo do tempo. A estrutura mais adequada para representar esses dados é um tensor de ordem 3.

As três dimensões do tensor, que também são conhecidas como modos, são definidas da seguinte forma: (1) o primeiro modo representa os pares usuário-dia (UDs), (2) o segundo modo representa os minutos e (3) o terceiro modo representa o tipo de tráfego (download ou upload). Os modos (1), (2) e (3) são denotados pelos índices i, j e k, respectivamente.

Defini-se cada elemento do tensor como x_{ijk}, representando o tráfego k medido no par UD i no j-ésimo minuto do dia, onde $0 \leq i \leq I$, $0 \leq j \leq 1439$ e $0 \leq k \leq 1$. Fixa-se $k = 0$ para o tráfego de download e $k = 1$ para o tráfego de upload.

Como a metodologia adotada é geral, poderiam-se utilizar alternativamente outras métricas de interesse (e.g, atrasos e perdas) ou também considerar outros modos no tensor, como a localização geográfica dos usuários em caso de dados móveis. A escolha dos modos depende da aplicação e deve ser ajustada dependendo do conjunto de dados disponível e da interpretação e avaliação dos resultados obtidos.

Passo 2: *análise de fatores*

Em seguida, utiliza-se o método de decomposição tensorial PARAFAC para a extração de características relevantes das séries de tráfego. Com base na Equação 3.1.1, ao aplicar-se o método PARAFAC no tensor definido no **Passo 1**, obtêm-se as variáveis a_{ir}, b_{jr} e c_{kr}. Elas representam as cargas do fator r correspondentes a UD i, minuto j e tipo de tráfego k, respectivamente.

A Figura 4.3 exemplifica a decomposição pelo PARAFAC aplicado ao conjunto de dados de tráfego coletado. São definidos quatro fatores ($R = 4$), cada um deles com um nome dado a partir da interpretação dos resultados do modelo. Mais detalhes são apresentados no Capítulo 5.

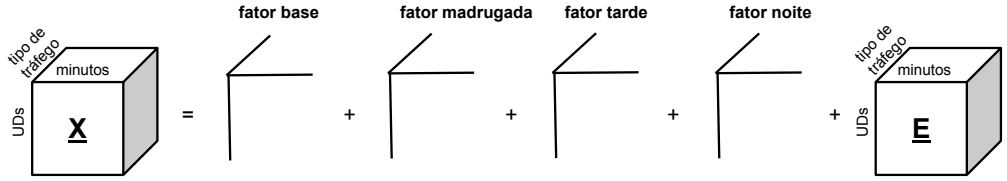

Figura 4.3: PARAFAC aplicado ao conjunto de dados de tráfego

Passo 3: *clusterização*

O próximo passo da metodologia é o agrupamento das séries temporais de tráfego com características temporais semelhantes. As caracterísisticas são representadas pelos *loadings* dos UDs obtidos a partir da análise fatorial. Um dos resultados da decomposição PARAFAC é o vetor de cargas $\mathbf{a_i} = (a_{i1}, \ldots, a_{iR})$ para cada UD i, onde $0 \leq i \leq I$ e $1 \leq r \leq R$. Portanto, o número de variáveis usadas na clusterização é igual ao número de fatores R do modelo.

O número de *clusters* é determindo avaliando-se o dendograma gerado pela clusterização hierárquica. Deve-se escolher um nível intermediário que maximize a similaridade das séries dentro dos grupos e que minimize a similaridade entre os grupos. Na prática, essa tarefa geralmente não é óbvia e exige uma escolha subjetiva que depende em algum grau dos objetivos da análise [61].

Passo 4: *análise de comportamento do usuário*

O último passo da metodologia consiste em obter a sequência de perfis de tráfego para cada usuário durante todo o período considerado. Para isso, utiliza-se um HMM com o objetivo de modelar o padrão de tráfego dos usuários.

Uma observação da HMM é definida como o *cluster* ao qual um usuário pertence em um determinado dia. Logo uma sequência de observações corresponde a sequência de perfis de tráfego (*clusters*) de um usuário ao longo do tempo. Dessa forma,

B é a distribuição de probabilidade do perfil de tráfego do usuário, podendo estar associada a cada um dos estados em **S**. Em outras palavras, $B(i,k)$ é a probabilidade do usuário pertencer ao *cluster* k condicionada ao estado S_i.

A partir das sequências de perfis diários de tráfego obtidas com os dados reais de todos os usuários do conjunto de dados, estima-se os parâmetros do modelo HMM usando o algoritmo *Baum-Welch*. O modelo HMM final obtido possui diversas aplicações. Pode ser usado no planejamento de capacidade e gerenciamento da rede. Além disso, novas sequências de perfis de usuários (observações) podem ser geradas a partir do modelo permitindo fazer previsões futuras de carga da rede.

Dados novos: *cálculo dos loadings e classificação*

A metodologia também permite a classificação de novas séries com uma árvore de decisão construída a partir dos *clusters* obtidos pelo conjunto de UDs inicial. Para isso deve-se calcular os *loadings* das novas séries.

O cálculo da carga de um novo par UD_κ, definida como $\tilde{a}_{\kappa r}$, $r = 1, ..., R$, é realizado aplicando-se o método PARAFAC novamente, conforme a Equação 3.1.1. Na equação, os valores b_{jr} e c_{kr} são os mesmos obtidos para o conjunto de UDs inicial e o valor $\tilde{a}_{\kappa r}$ é computado de forma a minimizar os erros quadráticos.

A árvore de decisão classifica um novo par UD pelo vetor $\tilde{a}_\kappa$, ordenando-o do nó raiz para algum nó folha. Cada nó interno da árvore especifica um teste de algum dos fatores r do novo UD e cada ramificação que desce a partir desse nó corresponde a um dos valores possíveis para esse fator.

Com base na classificação pode-se incluir novas sequências de *clusters* na análise, sem a necessidade de repetir as técnicas descritas nos passos anteriores. Isso pode facilitar o rastreamento e agilizar a detecção de mudanças no comportamento dos usuários.

Capítulo 5

Descobrindo perfis de tráfego

Este capítulo tem como principal objetivo apresentar os resultados obtidos através da aplicação da metodologia proposta no Capítulo 4. Aqui são detalhados os modelos desenvolvidos para o descobrimento e a caracterização de perfis de tráfego dos usuários residenciais.

Na Seção 5.1 são discutidas particularidades e estatísticas gerais do conjunto de dados, bem como as técnicas de preprocessamento implementadas antes de aplicar os métodos de análise. Em seguida, na Seção 5.2 são apresentados os resultados obtidos com a aplicação do método PARAFAC e da clusterização. O objetivo é caracterizar os perfis diários das amostras. Por fim, a Seção 5.3 faz uma análise dos perfis de comportamento dos usuários detectados durante todo o período de coleta com base nos perfis diários encontrados. São apresentados os resultados obtidos pela árvore de decisão e pelo Modelo de Markov Oculto (HMM).

5.1 Conjunto de dados e pré-processamento

O conjunto de dados deste trabalho considera amostras de tráfego coletadas durante quatro semanas (28 dias), de 20 de agosto a 16 de setembro de 2018. Uma amostra de tráfego de download (upload) representa o total de bytes recebido (enviado) a cada minuto por um determinado usuário em um determinado dia, conforme descrito na Seção 4. Denomina-se portanto uma série temporal como um par usuário-dia ou simplesmente um par UD.

A Figura 5.1 mostra a quantidade de roteadores ativos que realizaram a coleta do tráfego de download e de upload durante esse período, com os valores de antes e depois da aplicação de uma filtragem nos dados. O objetivo da filtragem é evitar a manipulação e análise de amostras de tráfego que possuem muitos espaços em branco (*missing data*).

Diversos motivos podem ocasionar espaços em branco nas amostras de tráfego. Tanto por problemas nos roteadores residenciais e/ou no servidor de coleta quanto

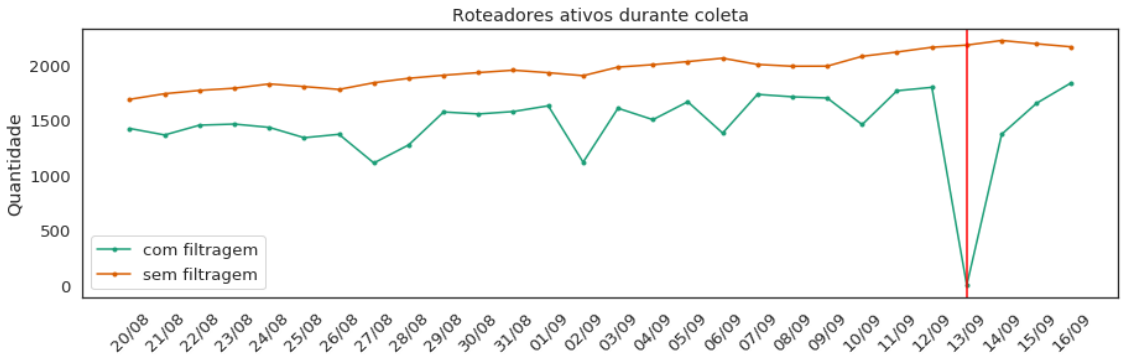

Figura 5.1: Quantidade de roteadores ativos durante o período de coleta

por algum bloqueio na comunicação entre eles. Também é possível que alguns usuários desconectem os seus roteadores manualmente em períodos de viagem ou até mesmo quando não estão acessando a rede.

Como os métodos de modelagem geralmente são muito sensíveis à ausencia de dados, limitam-se as séries a um máximo de dez minutos consecutivos de espaços em branco. Assim, mantém-se um equilíbrio entre a quantidade de roteadores ativos analisados e a precisão das técnicas empregadas. Note que, em função desse filtro, não existem amostras (UDs) no dia 13 de setembro, provavelmente por algum erro no servidor de coleta de dados.

Após a filtragem foram avaliados um total de 2.219 roteadores domésticos diferentes durante todo o período. Pelos resultados obtidos, relatados nas seções subsequentes deste capítulo, acredita-se que essa quantidade é representativa e abrange perfis básicos de uso da Internet.

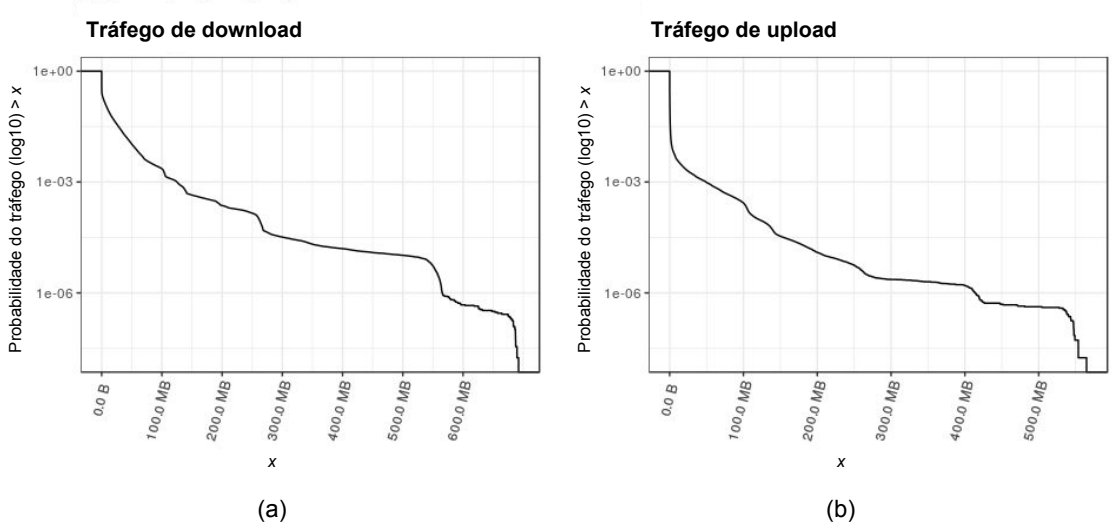

Figura 5.2: Função de Distribuição Complementar do tráfego

No que se segue, relatam-se algumas estatísticas do conjunto de dados. A Figura 5.2 apresenta a Função de Distribuição Complementar para cada tipo de tráfego

(i.e, download e upload) gerado a cada minuto pelos roteadores residenciais ativos. Escala logarítmica é usada no eixo y para enfatizar a diferença relativa.

Observe que uma cauda longa está presente, propriedade também identificada em trabalhos anteriores [8, 23]. Cerca de $0,3\%$ dos intervalos de amostragem de download (upload) usam mais de 100 MB (50 MB) por minuto (equivalente a taxas de 13 Mbps e 6.7 Mbps de download e upload, respectivamente) - ou seja, essas amostras representam minutos de uso intenso. A maioria dos intervalos medidos faz um uso menor da rede. Assim, apesar da baixa utilização da banda disponível nas redes de acesso, percebe-se a presença de tráfego em rajadas nas séries temporais.

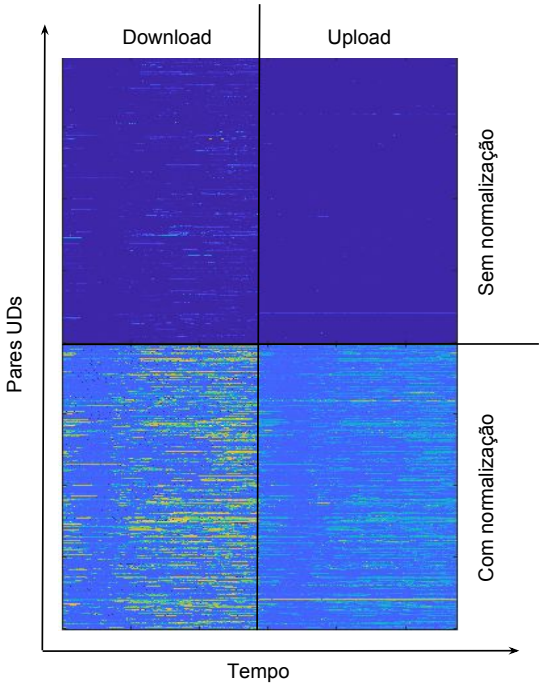

Figura 5.3: Antes e depois da tranformação logarítmica

Em função disso, a análise fatorial é precedida por uma normalização do tráfego, em que todas as amostras são consideradas em escala logarítmica. O objetivo dessa transformação é reduzir o efeito de rajadas e *outliers* nos resultados do modelo. A Figura 5.3 apresenta um *heatmap* de alguns UDs antes e depois da transformação logarítmica. A intensidade de cor indica o volume do tráfego utilizado por um UD em algum minuto do dia. Quanto mais escuro, menor a quantidade de bytes.

Antes da normalização, com a presença de algumas poucas rajadas muito intensas, a maior parte do tráfego medido está em azul escuro. Isso porque a diferença do valor entre as rajadas e o tráfego comum é muito alta. Após a normalização, as rajadas intensas permanecem porém com menor destaque. Assim, os detalhes do tráfego comum são mais visíveis e perceptíveis pelos métodos de análise empregados.

Outra característica interessante das amostras de tráfego são os padrões periódicos diários. Esses padrões podem ser identificados no gráfico na Figura 5.4. Ele

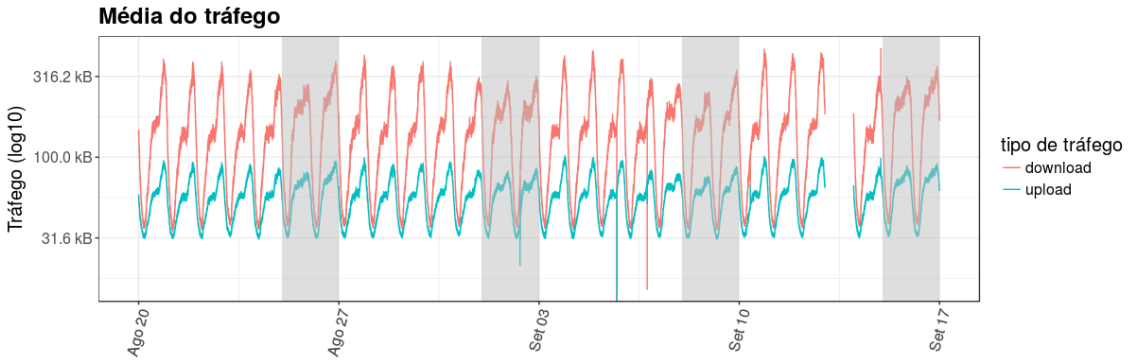

Figura 5.4: Média do tráfego de download e upload por minuto

mostra a média do tráfego de download e upload por minuto durante o período da coleta. A quantidade de amostras é igual a quantidade de roteadores ativos em cada um dos dias coletados, conforme apresenta a Figura 5.1. Veja que os dois tipos de tráfego possuem um comportamento similar, apesar de existir uma grande diferença entre a quantidade de bytes baixados e enviados.

Os picos no gráfico representam os momentos do dia com maior uso de rede, o que sugere que há um forte efeito diário. Além disso, os finais de semana, destacados em cinza, possuem um padrão diferente. Note portanto que a separação em dias das amostras (UDs) é feita com o objetivo de identificar essas diferenças.

As amostras são dividas em um grupo de 31.017 UDs e outro de 9.890 UDs para os conjuntos de treinamento e teste, respectivamente, totalizando 40.907 UDs. Das quatro semanas do conjunto de dados, o conjunto de treinamento é composto pelas três primeiras. Ele é utilizado para treinar o modelo PARAFAC e obter grupos que representam o perfil diário dos usuários residenciais. Esse processo é descrito em seguida, na Seção 5.2.

A última semana compõe o conjunto de teste. Esse conjunto é utilizado para examinar o algoritmo de classificação e, em conjunto com o conjunto de treinamento, gerar o HMM para caracterizar o comportamento dos usuários em períodos maiores que um dia. A Seção 5.3 apresenta os resultados desse processo.

5.2 Aprendendo perfis diários de tráfego

Inicialmente aplica-se o método PARAFAC para aprender de perfis diários de tráfego. O modelo é validado com no máximo quatro fatores ($R = 4$) utilizando-se o *Split-Half Validation* (SV). Os testes de similaridade do SV ocorrem entre os modelos PARAFAC dos subconjuntos independentes $(G_1 + G_2)$ vs. $(G_3 + G_4)$ e $(G_1 + G_3)$ vs. $(G_2 + G_4)$, conforme demonstra a Figura 5.5. Pelo menos um dos testes deve apresentar similaridade. O *Tucker Congruence Coefficient* (TCC) é superior a $0,95$ para os vetores de cargas entre os modelos $(G_1 + G_3)$ vs. $(G_2 + G_4)$.

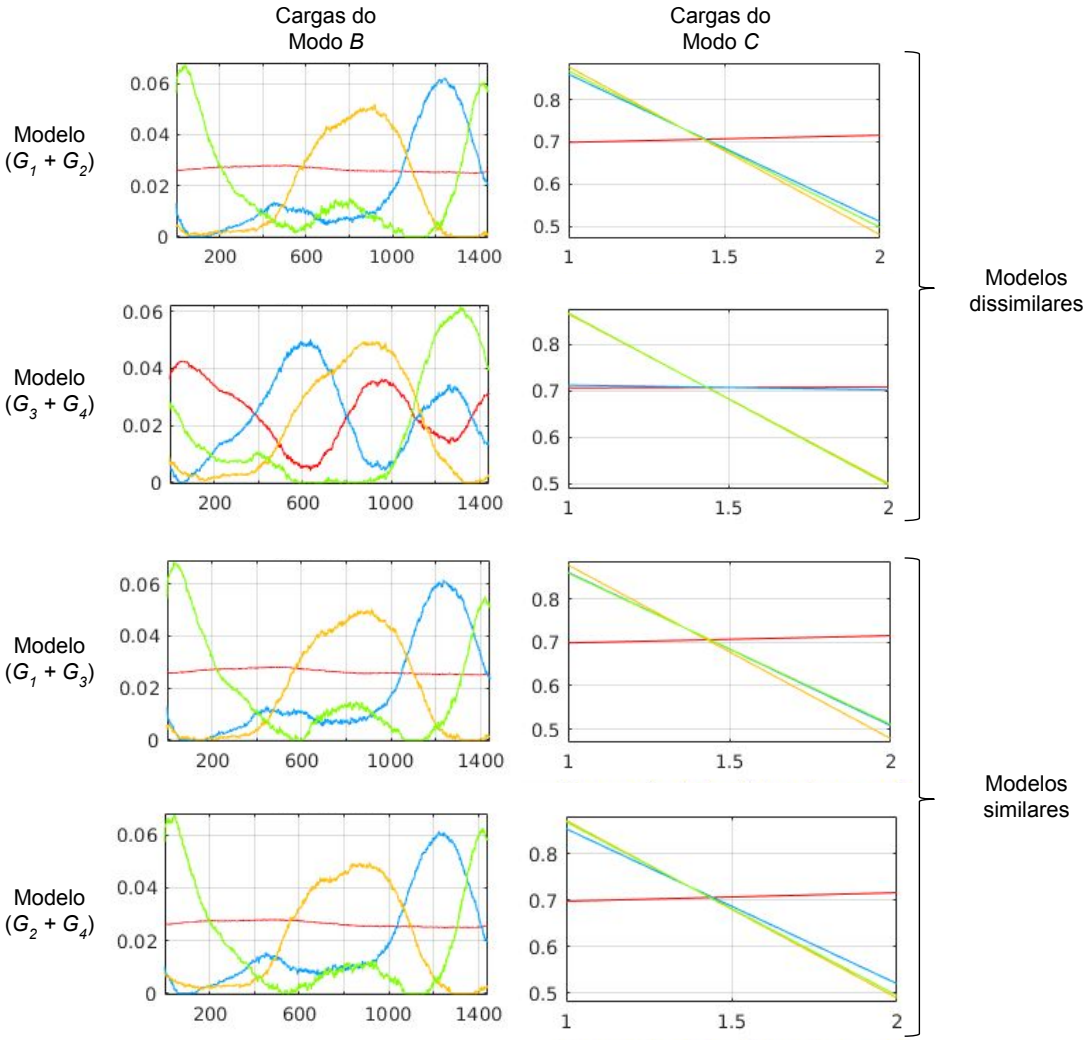

Figura 5.5: *Split-Half Validation* aplicado ao conjunto de treinamento

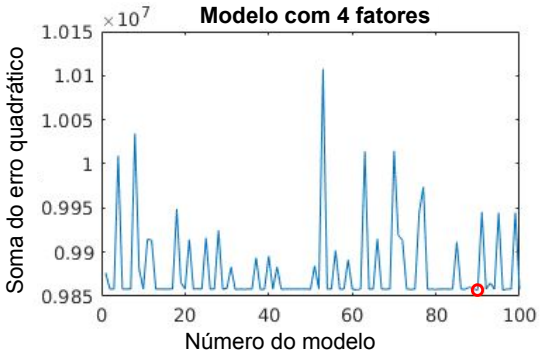

Figura 5.6: Inicializações randômicas do PARAFAC

O algoritmo usado para obter a solução do modelo PARAFAC é o método dos mínimos quadrados alternantes (ALS). Caso existam espaços em branco, eles são inicialmente substituídos pela média do tráfego de todo o tensor e são reestimados durante as iterações do ALS. O ALS é iterativo e para quando não ocorrem mu-

danças nas estimativas ou quando a diferença relativa da variância explicada entre duas iterações sucessivas está abaixo de um certo limite, conhecido como critério de convêrgência. O critério de convergência escolhido é 1e-5.

Para obtenção de um resultado confiável, são realizadas 100 inicializações randômicas do modelo (veja Figura 5.6). O modelo escolhido (círculo vermelho da figura) é aquele cuja soma do erro quadrático possui o menor valor; a sua porcentagem de variância explicada é igual a $98,55\%$.

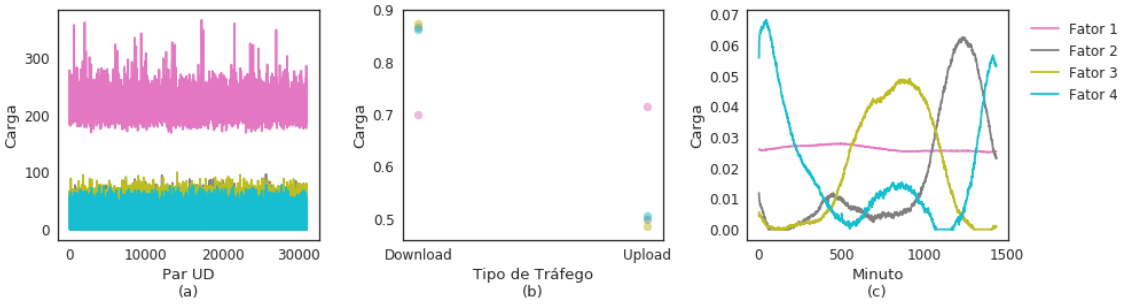

Figura 5.7: PARAFAC aplicado ao conjunto de dados de treinamento

A Figura 5.7 ilustra o modelo final da análise fatorial. Os quatro valores das cargas (*loadings*), um para cada fator e para cada uma das UDs modeladas estão na Figura 5.7(a). Esse resultado indica a intensidade do tráfego de download e upload dos UDs, isto é, para um determinado usuário-dia. Na Figura 5.7(b) têm-se os valores dos *loadings* para o tipo de tráfego e na Figura 5.7(c) são apresentadas os *loadings* de cada minuto para cada um dos quatro fatores no período de um dia.

Um dos fatores (de cor rosa no gráfico) nitidamente não está associado a horários de uso da rede para os tráfegos de download e upload. Os três fatores restantes estão associados ao uso mais intenso da rede em diferentes períodos do dia (madrugada (azul), tarde (verde) e noite (cinza)) e, apesar de serem presentes em ambos os tipos de tráfego, possuem valores mais altos para o download. A partir da observação dos valores dos *loadings* para um UD, é possível identificar o padrão de uso da rede deste UD. Por exemplo, se o valor do *loading* de um UD para o fator cinza for superior ao valor dos outros fatores, esta é uma indicação que esse usuário gera tráfego maior no período da noite para o dia correspondente ao par UD.

Para fins de comparação, a Figura 5.8 apresenta o modelo obtido pelo método bilinear *Singular Value Decomposition* (SVD) [40]. Como o SVD só pode ser executado em matrizes completas, implementa-se o algoritmo *SVDImpute*, proposto por Troyanskaya *et al.* [62]. Nesse algoritmo todos os espaços em branco são inicialmente substituídos pela média de cada coluna da matriz. O algoritmo funciona iterativamente até que a mudança dos valores estimados esteja abaixo de um certo limite. A estimativa é obtida pela combinação linear de r fatores mais significativos do modelo.

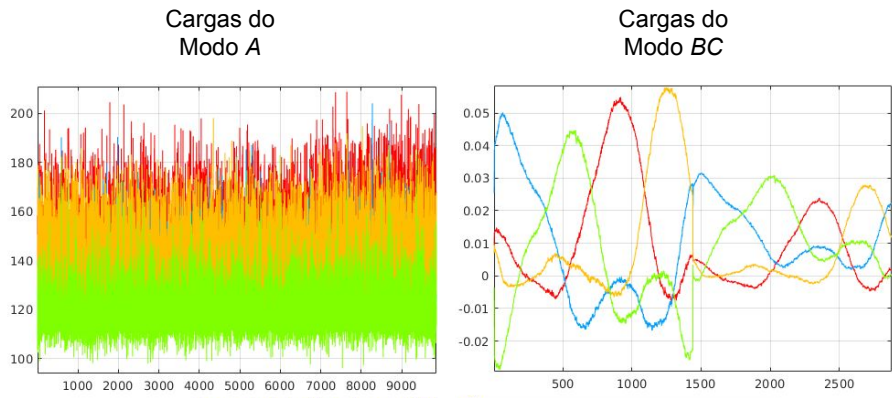

Figura 5.8: SVD aplicado ao conjunto de dados de treinamento

Aplica-se também o *Varimax* [14] para determinar o espaço fatorial mais adequado. A matriz decomposta possui dimensão $I \times JK$, ou seja, as linhas representam os UDs; as colunas entre 0 e 1439, o tráfego de download; e as colunas entre 1440 e 2879, o tráfego de upload. Conforme mencionado na Seção 3.1.2, um padrão diferente para cada um dos tipos de tráfego (i.e, fatias do modo C) é obtido (veja modo BC da Figura 5.8), mesmo quando possuem estruturas parecidas. O Parafac, por outro lado, apresenta um estrutura latente única para ambos os tipos de tráfego (modo C) ao longo do tempo (modo B).

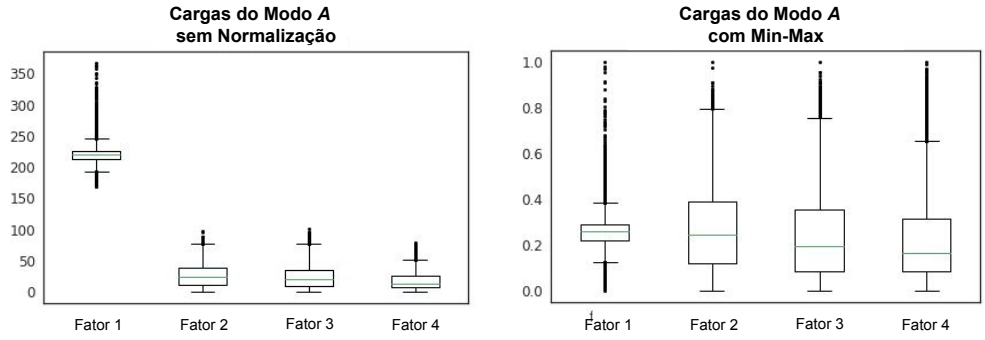

Figura 5.9: Normalização Min-Max nas cargas do modo A

Na próxima etapa, executa-se o algoritmo de clusterização hierárquica aglomerativa. A métrica de similaridade é a carga dos UDs para cada fator, ou seja, o vetor de cargas $\mathbf{a_i} = (a_{i1}, a_{i2}, a_{i3}, a_{i4})$ de cada UD_i. Devido aos valores das cargas variarem em até três ordens de grandeza, aplica-se a normalização Min-Max nas cargas do modo A. Essa normalização altera a escala das cargas para $[0, 1]$ com base nos valores mínimo e máximo de cada fator. A Figura 5.9 mostra o boxplot das cargas antes e depois da normalização Min-Max.

A partir do dendograma obtido, apresentado na Figura 5.10, são selecionados cinco *clusters*. A Figura 5.11 mostra a mediana do tráfego de download e de upload

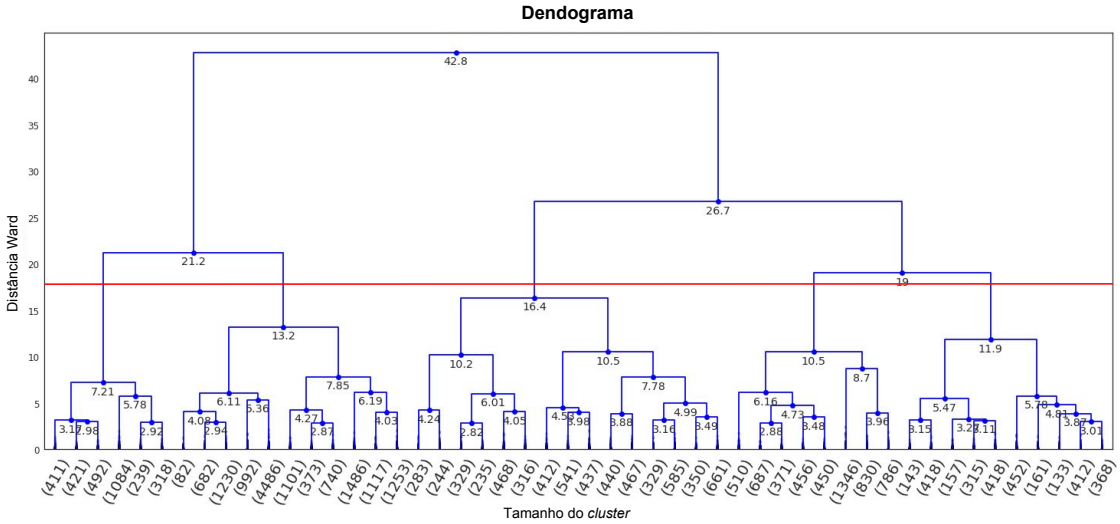

Figura 5.10: Dendograma obtido com a clusterização hierárquica aglomerativa

por minuto para todos os UDs de cada *cluster*. Nela aparecem tanto o resultado da clusterização pelo conjunto de treinamento quanto o resultado da classificação pelo conjunto de testes. Este último é explicado na próxima seção.

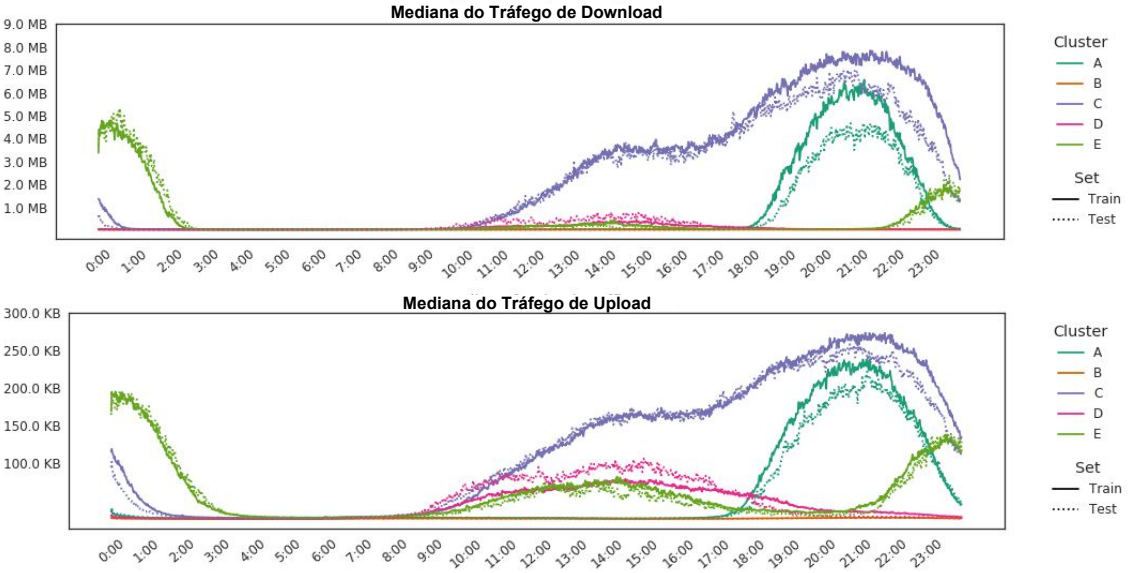

Figura 5.11: Mediana do tráfego de download e upload por minuto para todos os UDs de cada *cluster*: conjuntos de treinamento e de teste

Cada *cluster* representa um perfil de uso residencial da Internet. O *Cluster C* (roxo) agrupa UDs com o maior tráfego, concentrado entre a tarde e a noite. Em contraste, o *Cluster B* (laranja) é composto de UDs com o menor tráfego durante as 24h do dia. As UDs do *Cluster A* (verde escuro) são caracterizadas por uma alta demanda de banda à noite e, do *Cluster D* (rosa), das 10 às 18 horas. O *Cluster E* (verde claro) se distingue do *Cluster D* por apresentar uma demanda maior da rede durante a madrugada. Observe que o *Cluster E* apresenta padrão temporal similar

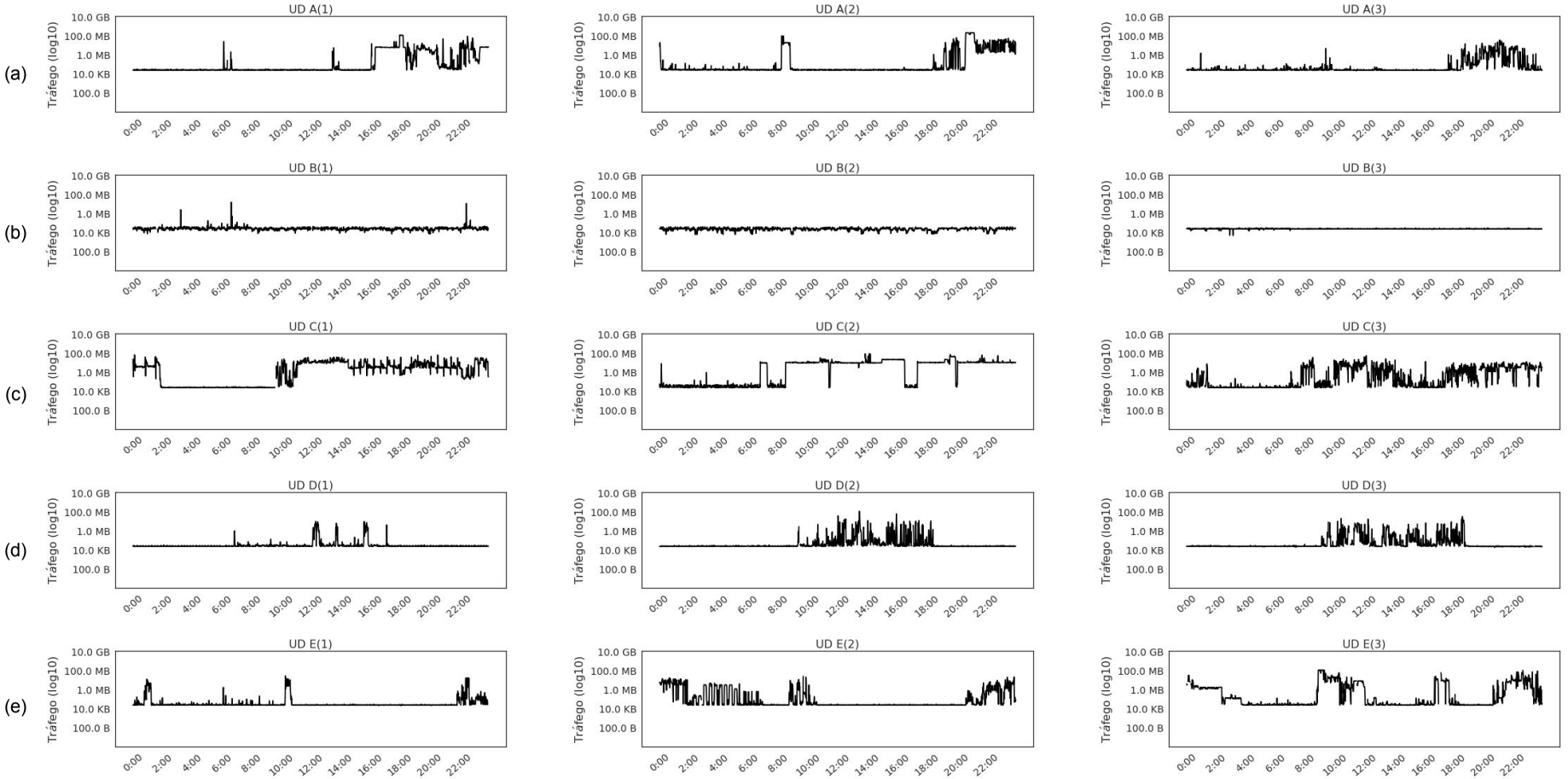

Figura 5.12: Tráfego de download de pares usuário-dia representativos de cada perfil: (a) UDs pertencentes ao *Cluster A*, (b) UDs pertencentes ao *Cluster B*, (c) UDs pertencentes ao *Cluster C*, (d) UDs pertencentes ao *Cluster D*, (e) UDs pertencentes ao *Cluster E*

ao Fator 4 do modo minuto ilustrado na Figura 5.7(c).

O agrupamento derivado das cargas do PARAFAC caracteriza perfis diários de tráfego distintos dos usuários que fazem parte do conjunto de treinamento. A Figura 5.12 apresenta três exemplos representativos do tráfego de download dos UDs pertencentes a cada um dos perfis diários encontrados. Note que na Figura 5.12(a) os pares UD apresentam um maior volume de tráfego à noite. Os UDs na Figura 5.12(b) apresentam um tráfego baixo e aproximadamente constante ao longo das 24h. Já os UDs na Figura 5.12(c) apresentam maior tráfego no período da tarde/noite. Os UDs da Figura 5.12(d) geram maior volume de tráfego no período da manhã/tarde, e os UDs da Figura 5.12(e) na madrugada/noite. Esses exemplos ilustram a similaridade dos perfis de tráfego de pares UD pertences ao mesmo cluster.

Por fim, é importante notar que existe correlação temporal entre o tráfego de download e upload. Se o valor da mediana do tráfego de download para um determinado *cluster*/período do dia é alto, o valor da mediana do tráfego de upload para esse mesmo *cluster*/período do dia também será alto.

5.3 Análise do comportamento de tráfego dos usuários

Um dos objetivos deste trabalho é classificar um novo UD em um dos perfis definidos pela análise fatorial e clusterização. O primeiro passo é calcular as cargas do modo UD para esse novo par UD_κ utilizando o PARAFAC. Em seguida, classifica-se o novo UD pelo vetor de cargas obtido, $\tilde{\boldsymbol{a}}_{\boldsymbol{\kappa}}$, seguindo o caminho de decisão proposto pela árvore ilustrada na Figura 5.13.

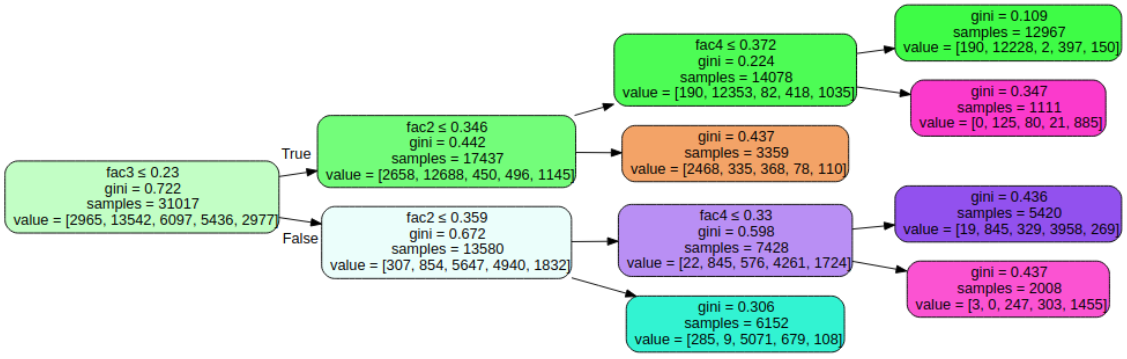

Figura 5.13: Árvore de decisão obtida a partir do conjunto inicial de UDs.

A árvore é podada de modo a alcançar uma configuração generalizável para a classificação de novos UDs (do conjunto de teste), sem realizar *overfitting* nos dados. Para isso, defini-se um valor mínimo para diminuição da impureza de 0.02. Assim, um nó só é dividido se essa divisão induzir uma diminuição da impureza maior ou

igual a esse valor. Após a poda, a precisão da árvore é de 84% para o conjunto de treinamento. A quantidade de UDs em cada *cluster* de ambos os conjuntos estão organizados na Tabela 5.1.

Tabela 5.1: Quandidade de UDs em cada cluster

	Cluster A	*Cluster B*	*Cluster C*	*Cluster D*	*Cluster E*
Treinamento	2.965	13.542	6.097	5.436	2.977
Teste	1.649	3.637	2.462	1.312	830

Como a análise é **não supervisionada**, verifica-se o resultado da classificação de forma visual pela Figura 5.11. As linhas tracejadas mostram a mediana do tráfego de download e de upload por minuto para as séries de teste classificadas. A mediana do tráfego para ambos os conjuntos de treinamento e teste é muito semelhante ao longo do dia. Resultados parecidos foram obtidos usando outros algoritmos de classificação como aquele que associa cada UD ao *cluster* de centróide mais próximo (i.e, *Nearest Centroid*) ou pela regra dos k vizinhos mais próximos (i.e, *k-Nearest Neighbor*).

É interessante observar como o perfil diário dos usuários evolui ao longo dos 28 dias. Na Figura 5.14 é apresentado o percentual de usuários pertencentes a cada um dos *clusters* para cada dia. Para facilitar a interpretação, é disposto no gráfico um fundo cinza nos finais de semana. O *Cluster B* é predominante em qualquer um dos dias, indicando que aproximadamente 45% dos usuários gera pouco tráfego na rede em qualquer período do dia.

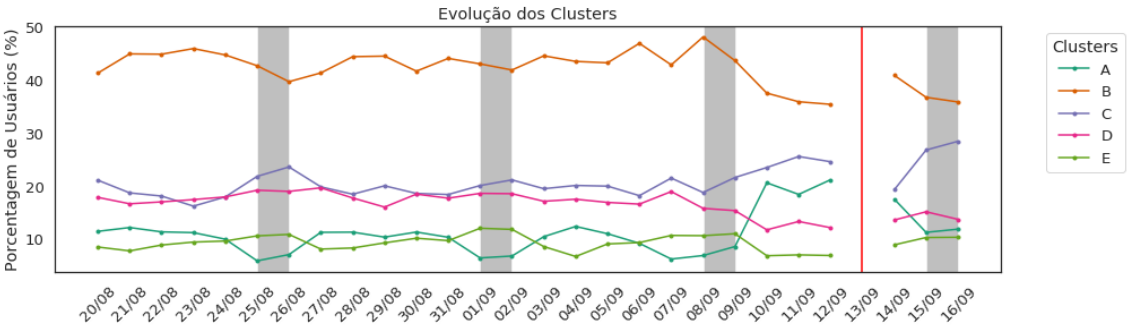

Figura 5.14: Evolução dos *clusters* ao longo dos 28 dias

Além disso, pode-se observar um comportamento diferenciado entre os dias da semana e os finais de semana. A porcentagem de usuários pertencentes ao *Cluster A* (uso maior da rede à noite) durante a semana diminui ao mesmo tempo que a porcentagem de usuários no *Cluster E* (uso maior da rede na madrugada) aumenta durante os finais de semana e no feriado do dia 7 de setembro. Em geral, os usuários que trabalham durante a semana só conseguem acessar a rede doméstica durante a noite, entre 19h e 23h. Nos finais de semana, por outro lado, uma quantidade

maior de usuários passa a utilizar a Internet de forma mais intensa a partir das 23h. No entanto, não são necessariamente os mesmos usuários que migram do *Cluster A* para o *Cluster E*.

Por fim, existe uma mudança de comportamento na última semana do conjunto de dados. O acesso a rede parece se tornar mais intenso, já que a proporção de usuários do *Cluster B* diminui e a de usuários dos *Clusters A* e *C* aumenta consideravelmente. Não se sabe ao certo que evento causou esse comportamento, mas pode ter relação com o atentado a um dos candidatos a presidência ocorrido no dia 06/09/2018, possivelmente pela procura por notícias imediatamente após o feriado.

Na última etapa do nosso trabalho caracterizam-se as sequências dos perfis diários dos usuários através de um HMM. É definido um modelo com cinco estados ocultos. As probabilidades de emissão de símbolos (B) são inicializadas aleatoriamente; a matriz de transição (A) e o vetor de probabilidades inicial ($\boldsymbol{\pi}$) são inicializados uniformemente. Em seguida, o algoritmo Baum-Welch é usado para refinar os parâmetros do modelo. Os parâmetros $\boldsymbol{\pi}$ e A estimados a partir dos dados reais estão na Tabela 5.2. A Figura 5.15 apresenta B, ou seja, a distribuição de probabilidade do perfil diário do usuário associada a cada estado oculto S_i, onde $i = 1, ..., 5$.

Tabela 5.2: Modelo de Markov oculto com cinco estados

de/para	$s0$	$s1$	$s2$	$s3$	$s4$
início (π)	0,15	0,20	0,34	0,11	0,19
$s0$	0,97	0,02	0,01	0,00	0,00
$s1$	0,01	0,98	0,00	0,00	0,00
$s2$	0,01	0,00	0,97	0,01	0,01
$s3$	0,01	0,01	0,02	0,95	0,00
$s4$	0,01	0,01	0,01	0,01	0,97

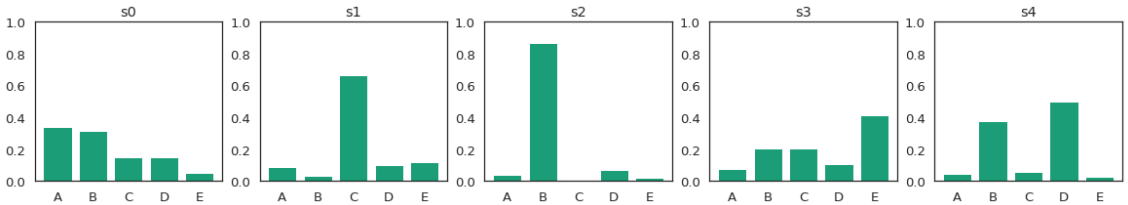

Figura 5.15: Distribuição de probabilidade do perfil diário para cada estado do HMM

A partir da Tabela 5.2, observa-se que os usuários costumam se manter com uma probabilidade alta no mesmo estado do HMM. Esse resultado indica que usuários

tendem a manter perfis diários de tráfego específicos que estão associados a um determinado estado do HMM. Por exemplo, usuários que se mantêm na maior parte do tempo no estado S_2 têm o seu perfil diário de tráfego melhor caracterizado pelo *Cluster* B, ou seja, são usuários que geram pouco tráfego. Já a maioria dos usuários que se mantêm na maior parte do tempo no estado S_1 geram um volume de tráfego alto durante a tarde e a noite. Esse tipo de resultado permite que os ISPs realizem um planejamento mais adequado das suas redes com base na provável demanda diária dos usuários.

Aplica-se o algoritmo de *Viterbi* para um pequeno subconjunto de usuários visando obter a sequência de estados mais prováveis do HMM, a partir dos perfis diários obtidos pela clusterização. Na Figura 5.16, cada linha corresponde a um usuário; as cores, ao estado do HMM; e, a letra dentro de cada quadrado, ao *cluster* ao qual o usuário está associado naquele dia. Pode-se notar que os três primeiros usuários se mantêm no mesmo estado do HMM durante as 4 semanas. O primeiro usuário, por exemplo, se mantém no estado S_4 durante os 28 dias. No estado S_4 existe uma alta probabilidade do perfil diário de tráfego ser o associado aos *Clusters* B e D, indicando um usuário que gera pouco tráfego na rede. Outros usuários podem mudar de estado ao longo do tempo (quarta e quinta linhas da Figura 5.16).

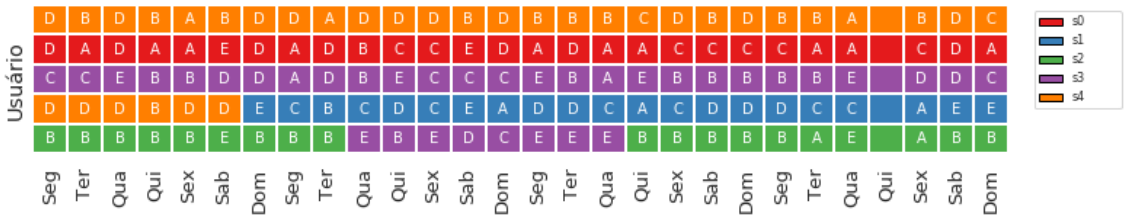

Figura 5.16: Sequência de estados do HMM e perfil diário de tráfego para um grupo de usuários.

Capítulo 6

Conclusões

Este trabalho propôe um *framework* simples para detectar estruturas temporais e padrões comportamentais da atividade de tráfego de usuários residenciais. O *framework* é composto por um conjunto de técnicas não supervisionadas de aprendizado de máquina. Dessa forma, mostra-se como é possível extrair com eficiência características relevantes do conjunto de dados, sem pré-rotular os dados e preservando a privacidade dos usuários.

Em suma, apresenta-se um modelo do perfil diário de usuários residenciais com base nas séries temporais de download e upload. Para isso, utiliza-se uma técnica de decomposição de tensores (PARAFAC) para capturar fatores interpretáveis e intrínsicos ao conjunto de dados. Esses fatores sugerem padrões diários comuns de tráfego ao longo do dia.

Também propõe-se um modelo do perfil de comportamento de usuários residenciais em períodos maiores que um dia. Elabora-se um modelo de Markov oculto (HMM) a partir das sequencias de perfis diários dos usuários obtidas a partir do modelo PARAFAC. O modelo final obtido indica que os usuários tendem a manter um padrão específico ao longo do tempo.

Uma das principais aplicações deste trabalho é apoiar as tarefas de planejamento de capacidade e gerenciamento da rede. Uma maneira de melhorar a utilização da rede, por exemplo, é mesclar no mesmo conjunto de recursos da rede os usuários que têm um perfil de tráfego contrastante ao longo dos dias. Ou seja, usuários cujos períodos de maior utilização estão em intervalos de tempo separados. Isso pode liberar alguns recursos do Provedor de Serviço Internet (ISP).

Dentre as possíveis aplicações adicionais deste trabalho citam-se, por exemplo: (a) o uso do modelo HMM (generativo) em simulações e estudo de cenários para avaliar o impacto na rede devido ao aumento de usuários de determinados perfis; (b) relacionar a topologia da rede aos perfis de tráfego identificados a fim de se estudar possíveis correlações com outras medidas de desempenho, como perda e latência.

Referências Bibliográficas

[1] FUMO, A., FIORE, M., STANICA, R. "Joint spatial and temporal classification of mobile traffic demands". In: *INFOCOM*, pp. 1–9. IEEE, 2017.

[2] KIM, J., HWANG, J., KIM, K. "High-performance internet traffic classification using a Markov model and Kullback-Leibler divergence", *Mobile Information Systems*, 2016.

[3] MORICHETTA, A., MELLIA, M. "LENTA: Longitudinal Exploration for Network Traffic Analysis". In: *ITC*, 2018.

[4] NGUYEN, T. T., ARMITAGE, G. "A survey of techniques for internet traffic classification using machine learning", *IEEE Communications Surveys & Tutorials*, v. 10, n. 4, pp. 56–76, 2008.

[5] SOYSAL, M., SCHMIDT, E. G. "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison", *Performance Evaluation*, v. 67, n. 6, pp. 451–467, 2010.

[6] WRIGHT, C., MONROSE, F., MASSON, G. M. "HMM profiles for network traffic classification". In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 9–15. ACM, 2004.

[7] TREVISAN, M., GIORDANO, D., DRAGO, I., et al. "Five Years at the Edge: Watching Internet from the ISP Network". In: *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '18, pp. 1–12, 2018. ISBN: 978-1-4503-6080-7. doi: 10.1145/3281411.3281433. Disponível em: <`http://doi.acm.org/10.1145/3281411.3281433`>.

[8] CROVELLA, M., KRISHNAMURTHY, B. *Internet measurement: infrastructure, traffic and applications*. John Wiley & Sons, Inc., 2006.

[9] KROONENBERG, P. M. *Three-mode principal component analysis: Theory and applications*, v. 2. DSWO press, 1983.

[10] SMILDE, A., BRO, R., GELADI, P. *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons, 2005.

[11] STEDMON, C. A., BRO, R. "Characterizing dissolved organic matter fluorescence with parallel factor analysis: a tutorial", *Limnology and Oceanography: Methods*, v. 6, n. 11, pp. 572–579, 2008.

[12] RABANSER, S., SHCHUR, O., GÜNNEMANN, S. "Introduction to Tensor Decompositions and their Applications in Machine Learning", *arXiv preprint arXiv:1711.10781*, 2017.

[13] 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.

[14] KAISER, H. F. "The varimax criterion for analytic rotation in factor analysis", *Psychometrika*, v. 23, n. 3, pp. 187–200, 1958.

[15] CALLADO, A., KAMIENSKI, C., SZABÓ, G., et al. "A survey on internet traffic identification", *IEEE communications surveys & tutorials*, v. 11, n. 3, 2009.

[16] LAKHINA, A., CROVELLA, M., DIOT, C. "Diagnosing network-wide traffic anomalies". In: *ACM SIGCOMM Computer Communication Review*, v. 34, pp. 219–230. ACM, 2004.

[17] LAKHINA, A., CROVELLA, M., DIOT, C. "Characterization of network-wide anomalies in traffic flows". In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 201–206. ACM, 2004.

[18] 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.

[19] SUN, J., TAO, D., FALOUTSOS, C. "Beyond streams and graphs: dynamic tensor analysis". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 374–383. ACM, 2006.

[20] MARUHASHI, K., GUO, F., FALOUTSOS, C. "Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis". In: *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pp. 203–210. IEEE, 2011.

[21] XIE, K., WANG, L., WANG, X., et al. "Accurate recovery of Internet traffic data: A tensor completion approach". In: *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*, pp. 1–9. IEEE, 2016.

[22] XIE, K., PENG, C., WANG, X., et al. "Accurate recovery of internet traffic data under dynamic measurements". In: *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pp. 1–9. IEEE, 2017.

[23] LELAND, W. E., WILLINGER, W., TAQQU, M. S., et al. "On the self-similar nature of ethernet traffic", *ACM SIGCOMM Computer Communication Review*, v. 25, n. 1, pp. 202–213, 1995.

[24] LOPEZ-MARTIN, M., CARRO, B., SANCHEZ-ESGUEVILLAS, A., et al. "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things", *IEEE Access*, v. 5, pp. 18042–18050, 2017.

[25] ACETO, G., CIUONZO, D., MONTIERI, A., et al. "Mobile encrypted traffic classification using deep learning". In: *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–8. IEEE, 2018.

[26] WAMSER, F., PRIES, R., STAEHLE, D., et al. "Traffic characterization of a residential wireless Internet access", *Telecommunication Systems*, v. 48, n. 1-2, pp. 5–17, 2011.

[27] VELAN, P., MEDKOVÁ, J., JIRSÍK, T., et al. "Network traffic characterisation using flow-based statistics". In: *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pp. 907–912. IEEE, 2016.

[28] GARCÍA-DORADO, J. L., FINAMORE, A., MELLIA, M., et al. "Characterization of isp traffic: Trends, user habits, and access technology impact", *IEEE Transactions on Network and Service Management*, v. 9, n. 2, pp. 142–155, 2012.

[29] XU, K., WANG, F., GU, L. "Behavior analysis of internet traffic via bipartite graphs and one-mode projections", *IEEE/ACM Transactions on Networking (TON)*, v. 22, n. 3, pp. 931–942, 2014.

[30] JIANG, H., GE, Z., JIN, S., et al. "Network prefix-level traffic profiling: Characterizing, modeling, and evaluation", *Computer Networks*, v. 54, n. 18, pp. 3327–3340, 2010.

[31] NOGUEIRA, A., DE OLIVEIRA, M. R., SALVADOR, P., et al. "Using neural networks to classify internet users". In: *null*, pp. 183–188. IEEE, 2005.

[32] NOGUEIRA, A., DE OLIVEIRA, M. R., SALVADOR, P., et al. "Classification of internet users using discriminant analysis and neural networks". In: *Next Generation Internet Networks, 2005*, pp. 341–348. IEEE, 2005.

[33] WEI, S., MIRKOVIC, J., KISSEL, E. "Profiling and Clustering Internet Hosts." *DMIN*, v. 6, pp. 269–75, 2006.

[34] GUO, M.-J., PENG, L., FANG, L., et al. "Analysis on preference patterns of ADSL users", *The Journal of China Universities of Posts and Telecommunications*, v. 19, pp. 73–79, 2012.

[35] NABOULSI, D., STANICA, R., FIORE, M. "Classifying call profiles in large-scale mobile traffic datasets". In: *INFOCOM, 2014 Proceedings IEEE*, pp. 1806–1814. IEEE, 2014.

[36] WANG, J., TANG, J., XU, Z., et al. "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach". In: *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pp. 1–9. IEEE, 2017.

[37] RABINER, L. R. "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, v. 77, n. 2, pp. 257–286, 1989.

[38] THURSTONE, L. L. "Multiple factor analysis." *Psychological Review*, v. 38, n. 5, pp. 406, 1931.

[39] JOLLIFFE, I. "Principal component analysis". In: *International encyclopedia of statistical science*, Springer, pp. 1094–1096, 2011.

[40] GOLUB, G. H., VAN LOAN, C. F. *Matrix computations*, v. 3. JHU Press, 2012.

[41] HARSHMAN, R. A. "Foundations of the PARAFAC procedure: Models and conditions for an"explanatory"multimodal factor analysis", 1970.

[42] HARSHMAN, R. A., LUNDY, M. E. "The PARAFAC model for three-way factor analysis and multidimensional scaling", *Research methods for multimode data analysis*, v. 46, pp. 122–215, 1984.

[43] KRUSKAL, J. "Multilinear methods". In: *Proc. Symp. Appl. Math*, v. 28, p. 75, 1983.

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

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

[46] 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.

[47] HARSHMAN, R. A., BERENBAUM, S. A. "Basic concepts underlying the PARAFAC-CANDECOMP three-way factor analysis model and its application to longitudinal data", *Present and past in middle life/edited by Dorothy H. Erchorin...[et al]*, 1981.

[48] ACAR, E., YENER, B. "Unsupervised multiway data analysis: A literature survey", *IEEE transactions on knowledge and data engineering*, v. 21, n. 1, pp. 6–20, 2009.

[49] KRUSKAL, J. B. "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics", *Linear algebra and its applications*, v. 18, n. 2, pp. 95–138, 1977.

[50] JAIN, A. K., MURTY, M. N., FLYNN, P. J. "Data clustering: a review", *ACM computing surveys (CSUR)*, v. 31, n. 3, pp. 264–323, 1999.

[51] WARD JR, J. H. "Hierarchical grouping to optimize an objective function", *Journal of the American statistical association*, v. 58, n. 301, pp. 236–244, 1963.

[52] LEGENDRE, P., LEGENDRE, L. "Numerical ecology. 3rd", *Elsevier*, 2012.

[53] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., et al. "Classification and decision trees", *Wadsworth, Belmont*, v. 378, 1984.

[54] E SILVA, E. D. S., 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*, Springer, pp. 112–128, 2011.

[55] GHAHRAMANI, Z. "An introduction to hidden Markov models and Bayesian networks", *International journal of pattern recognition and artificial intelligence*, v. 15, n. 01, pp. 9–42, 2001.

[56] BAUM, L. E., PETRIE, T., SOULES, G., et al. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains", *The annals of mathematical statistics*, v. 41, n. 1, pp. 164–171, 1970.

[57] FORNEY, G. D. "The viterbi algorithm", *Proceedings of the IEEE*, v. 61, n. 3, pp. 268–278, 1973.

[58] OPENWRT. "OpenWrt: Wireless Freedom". 2019. Disponível em: <`https://openwrt.org/`>. Acessado em 09/01/2019.

[59] SAMKNOWS. "SamKnows: Internet Performance Measurement". 2019. Disponível em: <`https://www.samknows.com/`>. Acessado em 09/01/2019.

[60] SUNDARESAN, S., BURNETT, S., FEAMSTER, N., et al. "BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks." In: *USENIX Annual Technical Conference*, pp. 383–394, 2014.

[61] WILKS, D. S. "Cluster analysis". In: *International geophysics*, v. 100, Elsevier, pp. 603–616, 2011.

[62] TROYANSKAYA, O., CANTOR, M., SHERLOCK, G., et al. "Missing value estimation methods for DNA microarrays", *Bioinformatics*, v. 17, n. 6, pp. 520–525, 2001.