Universidade Federal
do Rio de Janeiro
Escola Politécnica

DESCOBRINDO PERFIS DE USUÁRIOS DA INTERNET USANDO ANÁLISE NÃO SUPERVISIONADA

Anderson de Souza Barbosa

Projeto de Graduação apresentado ao Curso de Engenharia de Computação e Informação da Escola Politécnica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

Orientador: Rosa Maria Meri Leão

Rio de Janeiro
Março de 2019

DESCOBRINDO PERFIS DE USUÁRIOS DA INTERNET USANDO ANÁLISE NÃO SUPERVISIONADA

Anderson de Souza Barbosa

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE COMPUTAÇÃO E INFORMAÇÃO.

Examinadores:

Prof. Rosa Maria Meri Leão, Dr.

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

Prof. Daniel Sadoc Menasche, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2019

Barbosa, Anderson de Souza

Descobrindo Perfis de Usuários da Internet Usando Análise Não Supervisionada/Anderson de Souza Barbosa. – Rio de Janeiro: UFRJ/POLI – COPPE, 2019.

XII, 40 p.: il.; 29,7 cm.

Orientador: Rosa Maria Meri Leão

Projeto (graduação) – UFRJ/ Escola Politécnica/ Curso de Engenharia de Computação e Informação, 2019.

Referências Bibliográficas: p. 39 – 40.

1. Análise Não Supervisionada. 2. Medições em Redes. 3. Perfis de Usuários da Internet. 4. PARAFAC. 5. Clusterização. I. Leão, Rosa Maria Meri. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Computação e Informação. III. Título.

Dedico esta, bem como todas as minhas demais conquistas, aos meus amados pais, Leci e Jorgito.

Agradecimentos

Primeiramente, agradeço a Deus por ter me dado saúde e força para superar todas as dificuldades ao longo da minha graduação.

Agradeço aos meus pais, Leci e Jorgito, por todo o amor, apoio e incentivo que recebi ao longo de toda a minha vida e que resultam em mais esta conquista.

Agradeço à professora Rosa pela oportunidade de fazer iniciação científica no laboratório LAND e pela orientação ao longo deste trabalho.

Por fim, agradeço a todos os demais que direta ou indiretamente contribuíram para a realização deste trabalho.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

DESCOBRINDO PERFIS DE USUÁRIOS DA INTERNET USANDO ANÁLISE NÃO SUPERVISIONADA

Anderson de Souza Barbosa

Março/2019

Orientador: Rosa Maria Meri Leão

Curso: Engenharia de Computação e Informação

A complexidade da Internet está cada vez maior, tudo graças ao crescente número de serviços e ao crescimento da Internet das Coisas. Com isso, o comportamento dos usuários tem ficado cada vez mais diversificado, fazendo com que fique mais difícil entender o que acontece na rede e fazer previsões. É importante para um Provedor de Serviços de Internet (ISP) entender o comportamento de seus usuários para fazer um gerenciamento de rede mais eficiente. Para tentar ajudar os ISPs nessa tarefa, diversos trabalhos tem empregado técnicas de aprendizado de máquina com o objetivo de classificar os diferentes padrões de tráfego na Internet, o que pode ser usado para fazer previsões de tráfego futuro, por exemplo. Com uma proposta parecida, usamos dados coletados de roteadores residenciais em parceria com um ISP brasileiro para criar um modelo de comportamento de usuário. Ao contrário da maioria dos trabalhos neste tema, que utilizam apenas dados de tráfego, este trabalho também utiliza dados de atraso de rede e perda de pacotes. Primeiramente, utilizamos um método de análise de fatores para modelar os dados e, a partir do modelo obtido, utilizamos uma técnica de clusterização para encontrar grupos de usuários com diferentes perfis.

Palavras-chave: Análise Não Supervisionada, Medições em Redes, Perfis de Usuários da Internet, PARAFAC, Clusterização

Abstract of the Undergraduate Project presented to Poli/COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Computer and Information Engineer.

DISCOVERING INTERNET USER PROFILES USING UNSUPERVISED ANALYSIS

Anderson de Souza Barbosa

March/2019

Advisor: Rosa Maria Meri Leão

Course: Computer and Information Engineering

The complexity of the Internet is increasing, thanks to the growing number of services and the growth of the Internet of Things. As a result, user behavior has become increasingly diverse, making it harder to understand what is happening on the network and make predictions. It is important for an Internet Service Provider (ISP) to understand the behavior of its users to make network management more efficient. To try to assist ISPs in this task, several papers have employed machine-learning techniques to classify different traffic patterns on the Internet, which can be used to predict future traffic, for example. With a similar proposal, we use data collected from home routers in partnership with a Brazilian ISP to create a user behavior model. Unlike most works on this topic, which use only traffic data, this work also uses network delay and packet loss data. First, we used a factor analysis method to model the data and, from the obtained model, we used a clustering technique to find groups of users with different profiles.

Keywords: Unsupervised Analysis, Network Measurement, Internet User Profiles, PARAFAC, Clustering

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Internet atual é de grande complexidade e a tendência é que essa característica se torne cada vez mais intensa. Um dos fatores que mais colaboram para isso é o crescimento da Internet das Coisas, proporcionando um número cada vez maior de dispositivos conectados à Internet nas residências. Esse crescimento da complexidade da Internet traz grandes desafios aos Provedores de Serviço de Internet (ISPs), que precisam estar sempre melhorando suas estratégias de gerenciamento de rede.

Os ISPs tem usuários com perfis muito diferentes, com cada um usando a Internet de uma maneira específica. Há pessoas que usam a Internet principalmente para jogar on-line, aquelas que usam mais para acessar serviços de *streaming*, aquelas que usam apenas para acessar e-mails e redes sociais, além de outros perfis. Os usuários também se diferenciam pelas horas do dia e pelos dias da semana em que usam mais a Internet. É importante ter um modelo que permita entender as características dos diversos perfis de usuários para prever tráfego futuro, alocar recursos de forma mais eficiente e até mesmo detectar comportamentos anômalos [1].

A forma como as pessoas usam a Internet muda constantemente e isso vem acontecendo de maneira cada vez mais repentina [2]. Novos serviços surgem a todo momento, tendo mais opções de uso da Internet, os usuários mudam seus hábitos, tudo isso contribui para que o que se sabe sobre como a Internet é usada possa não ser mais verdade algum tempo depois. Este trabalho busca entender as características do uso atual da Internet do ponto de vista de um ISP. Acredita-se que este tipo de estudo deva ser realizado periodicamente pelos ISPs, para que eles possam identificar mudanças de comportamento de seus usuários e se adaptar à nova situação.

A maioria dos trabalhos que buscam classificar comportamentos de usuários da Internet analisam apenas dados de tráfego. Este trabalho, por outro lado, além dos dados de tráfego, também utiliza dados de atraso e de perda de pacotes, que podem trazer características importantes, como quais usuários possuem um desempenho de rede melhor. Com esses dados adicionais, os ISPs podem entender melhor o que acontece em suas redes.

Os dados utilizados neste trabalho foram coletados de roteadores domésticos em parceria com um ISP brasileiro. As métricas coletadas foram o tráfego de download, o tráfego de upload, a latência e a taxa de perda de pacotes. Dos dados coletados, foram selecionadas séries temporais de 647 usuários entre os dias 3 e 30 de setembro de 2018. Embora este estudo leve em consideração apenas um mês de dados, acredita-se que o seu resultado possa ser usado para entender o comportamento dos usuários em qualquer momento.

O objetivo deste trabalho é usar técnicas de aprendizado não supervisionado para detectar perfis de usuários com diferentes comportamentos. O primeiro passo foi criar um modelo dos dados usando análise de fatores. A partir desse modelo, é possível observar se, em um mês de dados, há um padrão que se repita diariamente e semanalmente. Os resultados mostram que há um padrão claro nos dados de tráfego. O segundo passo foi usar o modelo obtido para encontrar diferentes perfis de usuários através de uma técnica de clusterização. Foi possível ver que, para as três métricas, há diferenças grandes entre os grupos de usuários encontrados.

Além desta Introdução, esta monografia está organizada em mais 4 capítulos. O Capítulo 2 traz os conceitos básicos, explicando toda a teoria necessária para entender o restante deste trabalho. O Capítulo 3 traz a metodologia utilizada neste trabalho, mostrando o passo a passo de cada etapa. O Capítulo 4 traz todos os resultados obtidos, mostrando se foi possível obter perfis bem definidos de usuários. O Capítulo 5 traz a conclusão.

Capítulo 2

Conceitos Básicos

2.1 Tensores

Um tensor é um *array* multidimensional. Um tensor de primeira ordem é um vetor, um tensor de segunda ordem é uma matriz e um tensor de terceira ordem está ilustrado na Figura 2.1 [3]. Tensores são usados para representar dados que são muito complexos para serem representados e analisados através da tradicional abordagem bidimensional [4].

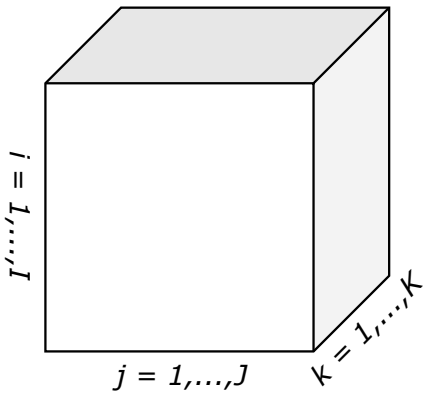


Figura 2.1: Um tensor de terceira ordem

2.1.1 Matricização

Um conceito importante para o entendimento da Seção 2.2 é a matricização, isto é, a transformação de tensores em matrizes. Na matricização, os elementos de um tensor são rearranjados de modo a formarem uma matriz. Assim, um tensor $3 \times 5 \times 4$, por exemplo, pode ser rearranjado para formar uma matriz 3×20, 5×12 ou 4×15. O número de diferentes maneiras de fazer a matricização é igual ao número de dimensões do tensor. Em uma matricização modo-n, o elemento $(i_1, i_2, ..., i_N)$ de um tensor é mapeado para o elemento (i_n, j) da matriz [3], onde

$$j = 1 + \sum_{\substack{k=1,\\ k\neq n}}^{N} (i_k - 1) J_k \quad \text{com} \quad J_k = \prod_{\substack{m=1,\\ m\neq n}}^{k-1} I_m \tag{2.1}$$

2.1.2 Tensores de posto 1

Um tensor $\underline{\mathbf{X}}$ de ordem N é de posto 1 se puder ser escrito como o produto diádico de N vetores [3], ou seja, se

$$\underline{\mathbf{X}} = a^{(1)} \otimes a^{(2)} \otimes \cdots \otimes a^{(N)} \tag{2.2}$$

onde "$\otimes$" representa o produto diádico.

2.2 PARAFAC

O PARAFAC (*Parallel Factors*) é um método de decomposição de tensores proposto por Harshman [5], que é uma generalização do *Principal Component Analysis* (PCA) [6] para dimensões maiores. O PARAFAC também foi proposto de forma independente por Carroll e Chang [7], que deram o nome de CANDECOMP (*Canonical Decomposition*). Este método decompõe um tensor em uma soma de tensores de posto 1. A decomposição de tensores tem como objetivo diminuir a dimensionalidade dos dados para tornar tratável um conjunto de dados que seja muito grande e também é um meio de detectar padrões nesses dados.

2.2.1 O modelo

Para tensores de três dimensões, que são aqueles nos quais estamos interessados, a decomposição gera três matrizes, $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$, que são chamadas de matrizes dos *loadings* (cargas). Considerando a decomposição de um tensor de três dimensões $\underline{\mathbf{X}}$ com dimensões $I \times J \times K$ com F componentes, a matriz $\mathbf{A}$ será $I \times F$, a $\mathbf{B}$ será $J \times F$ e a $\mathbf{C}$ será $K \times F$. Assim, uma maneira de representar o PARAFAC, elemento a elemento, é

$$x_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} c_{kf} + e_{ijk} \tag{2.3}$$

onde e_{ijk} representa um elemento do tensor de resíduos $\underline{\mathbf{E}}$. O modelo é encontrado minimizando a soma dos quadrados dos resíduos [8]. A Figura 2.2 ilustra a Equação 2.3. As três primeiras parcelas da soma são tensores de posto 1 obtidos através do produto diádico dos vetotes a, b e c correspondentes.

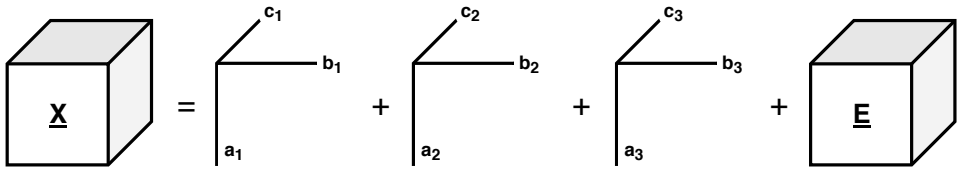


Figura 2.2: Representação gráfica de um modelo PARAFAC de 3 componentes

Uma outra forma de escrever o modelo é usando a matricização de $\underline{\mathbf{X}}$. Essa forma é útil pois o algoritmo usado para encontrar o modelo utiliza essa representação, como será visto mais a frente. O modelo na forma matricizada pode ser escrito de três diferentes formas

$$\mathbf{X}_{(1)} \approx \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T, \quad \mathbf{X}_{(2)} \approx \mathbf{B}(\mathbf{C} \odot \mathbf{A})^T, \quad \mathbf{X}_{(3)} \approx \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T \tag{2.4}$$

onde "$\odot$" representa o *Khatri-Rao product* [3] e $\mathbf{X}_{(n)}$ representa a matricização modo-n de $\underline{\mathbf{X}}$.

2.2.2 Posto de um tensor

O posto de um tensor é o menor número de tensores de posto 1 necessários em uma soma para gerá-lo. Assim, o posto de um tensor é o número de componentes necessários para uma decomposição PARAFAC exata, isto é, uma decomposição em que haja igualdade nas Equações 2.4. Embora a definição do posto de um tensor seja análoga à do posto de uma matriz, suas propriedades são bastante diferentes [3].

2.2.3 Implementação

O modelo PARAFAC pode ser encontrado pelo método *Alternating Least Squares* (ALS) assumindo sucessivamente que os *loadings* de dois modos são conhecidos e estimando o último modo a partir deles [8]. Essa estimativa é feita usando as Equações 2.4. Para este trabalho, foi utilizada uma implementação do PARAFAC contida na *N-Way Toolbox* para MATLAB [9].

Usando a primeira das Equações 2.4 como base para o ALS e considerando $\mathbf{Z} = \mathbf{C} \odot \mathbf{B}$, o modelo fica $\mathbf{X}_{(1)} = \mathbf{AZ}$. Assim, a estimativa de mínimos quadrados condicionais de $\mathbf{A}$ fica

$$\mathbf{A} = \mathbf{X}_{(1)}\mathbf{Z}^T(\mathbf{ZZ}^T)^{-1} \tag{2.5}$$

O pseudocódigo geral do algoritmo ALS para calcular o modelo PARAFAC é apresentado abaixo. O algoritmo recebe como entrada o tensor $\underline{\mathbf{X}}$ e o número de componentes, F:

1. Inicializar $\mathbf{B}$ e $\mathbf{C}$
2. Estimar $\mathbf{A}$ usando a Equação 2.5
3. Estimar $\mathbf{B}$ da mesma forma
4. Estimar $\mathbf{C}$ da mesma forma
5. Voltar ao passo 2 até que o algoritmo convirja, isto é, até que haja uma mudança nos *loadings* suficientemente pequena

2.2.4 Unicidade

A unicidade é uma propriedade que diz que as soluções do PARAFAC são frequentemente únicas [3]. Essa é uma das características que tornam o PARAFAC um método tão atrativo, pois métodos bilineares tem um problema com liberdade de rotação [8]. Kruskal [10, 11] usa o conceito de *k-rank* para estabelecer uma bem conhecida condição suficiente para a unicidade de uma solução. O *k-rank* de uma matriz é o maior valor k tal que quaisquer k colunas da matriz são linearmente independentes. Assim, para a decomposição de tensores de ordem 3, Kruskal [10, 11] mostra que para a unicidade é suficiente que

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2F + 2 \tag{2.6}$$

onde $k_{\mathbf{A}}$, $k_{\mathbf{B}}$ e $k_{\mathbf{C}}$ representam os *k-rank* das matrizes dos *loadings* do modelo e F representa o número de componentes.

2.2.5 Tucker's Congruence Coefficient

O *Tucker's Congruence Coefficient* (TCC) [12] é uma medida de proporcionalidade dos elementos em dois vetores. É calculado como o cosseno do ângulo entre esses dois vetores e é utilizado como um índice de similaridade entre fatores. Considerando dois fatores x e y com I elementos, o TCC entre eles é

$$\phi(x, y) = \frac{\sum_{i=1}^{I} x_i y_i}{\sqrt{\sum_{i=1}^{I} x_i^2 \sum_{i=1}^{I} y_i^2}} \tag{2.7}$$

2.2.6 Escolha do número de componentes

O trabalho [13] sugere um método chamado *Core Consistency Diagnostic* (CORCONDIA) para escolha do número ideal de componentes para modelar um tensor e o compara com a variação explicada. Um valor de CORCONDIA próximo de 100% indica que o modelo é adequado. Uma descrição detalhada de como o CORCONDIA é calculado pode ser encontrada em [13]. Assim como para o CORCONDIA, quanto maior a variação explicada, melhor, e ela é dada por

$$100\left(1-\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(x_{ijk}-m_{ijk})^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}x_{ijk}^2}\right) \tag{2.8}$$

onde x_{ijk} é um elemento do tensor original $\underline{\mathbf{X}}$ e m_{ijk} é o elemento correspondente do tensor $\underline{\mathbf{M}} = \underline{\mathbf{X}} - \underline{\mathbf{E}}$, onde $\underline{\mathbf{E}}$ é o tensor de resíduos .

2.3 Clusterização Hierárquica Aglomerativa

A clusterização é um método para agrupar elementos em clusters de forma que elementos que possuem determinadas características semelhantes fiquem no mesmo cluster. Algoritmos de clusterização podem ser de dois tipos de acordo com sua estrutura e operação: **aglomerativos** e **divisivos**. Os **algoritmos aglomerativos** começam com cada elemento em um cluster distinto e os clusters são agrupados sucessivamente até que um critério de parada seja alcançado. Já os **algoritmos divisivos** começam com todos os elementos em um único cluster e sucessivamente realiza divisões até que um critério de parada seja alcançado [14].

A clusterização hierárquica aglomerativa (CHA) é feita basicamente unindo sucessivamente os dois clusters mais próximos em um cluster maior. O passo a passo do algoritmo é apresentado abaixo:

1. Calcular as distâncias entre todos os pares de pontos e armazenar em uma matriz de distâncias
2. Colocar cada elemento em seu próprio cluster
3. Encontrar os clusters A e B mais próximos
4. Unir A e B em um novo cluster
5. Calcular a distância do novo cluster para os antigos e adicionar em uma nova matriz de distâncias
6. Voltar ao passo 3 até que todos os clusters tenham sido unidos em um único cluster

Há diversas maneiras de se calcular a distância entre dois clusters. Neste trabalho, será utilizada a média das distâncias entre os elementos dos clusters. Para executar a CHA, foi utilizado o pacote fastcluster para R [15]. Sejam A e B dois clusters, a distância entre eles é dada por

$$d(A, B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b) \tag{2.9}$$

onde $d(a, b)$ é a distância euclidiana entre os elementos a e b.

Capítulo 3

Metodologia

3.1 Coleta de Dados

Os dados foram coletados diretamente nos roteadores domésticos dos usuários do ISP parceiro. As métricas coletadas foram: tráfegos de download e upload dos usuários, latência e perda, que é a quantidade de pacotes perdidos em uma rajada de 100. Todas as métricas foram coletadas a cada minuto. A Figura 3.1 mostra como foi feita a coleta de dados. O roteador doméstico coleta os dados e envia para o servidor de coleta.

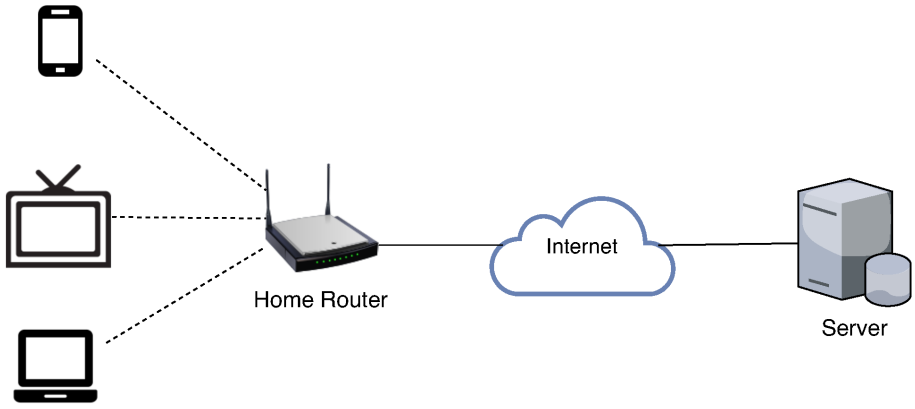
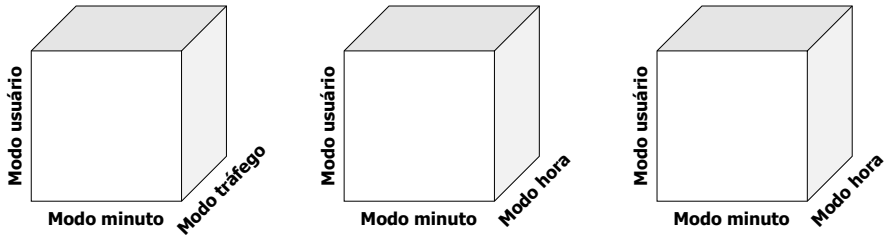


Figura 3.1: Esquema da coleta de dados

3.2 Análise de Fatores

Para este trabalho, foi definido que os dados a serem utilizados seriam aqueles coletados no período de 28 dias entre 3 e 30 de setembro de 2018. Foi definido também que seriam utilizadas apenas séries temporais que não possuíssem mais do que 50 valores ausentes consecutivos. No total, foram utilizadas 647 séries temporais de usuários.

Para proceder à análise de fatores, foram montados três tensores, um com os valores medidos dos tráfegos de download e upload, outro com os valores da latência e outro com os valores da perda. O primeiro tensor foi definido da seguinte forma: na primeira dimensão foi colocado o usuário ou série temporal, na segunda o minuto dentro do período de 28 dias e na terceira o tipo de tráfego. Os outros tensores foram definidos da seguinte forma: a primeira dimensão foi definida da mesma forma que no primeiro tensor, na segunda foi colocado o minuto da hora e na terceira a hora dentro do período de 28 dias. Assim, as dimensões do primeiro tensor são $647 \times 40320 \times 2$ e as dos outros dois são $647 \times 60 \times 672$. Os valores ausentes são preenchidos na *N-way Toolbox* usando o algoritmo *Expectation-Maximization* [9]. A Figura 3.2 mostra a estrutura dos tensores montados.



(a) Tensor dos tráfegos (b) Tensor da latência (c) Tensor da perda

Figura 3.2: Estrutura dos tensores montados

3.2.1 Validação

Para validar um modelo com determinado número de componentes, este trabalho utiliza a abordagem sugerida em [16]. que é utilizar a *Split-Half Validation* (SHV) [17] em conjunto com o *Tucker's Congruence Coefficient*. Nesta abordagem, o conjunto de dados é dividido em quatro grupos, 1, 2, 3 e 4. O grupo i fica com as amostras de número $i, i+4, i+8, i+12$ e assim por diante, para $i = 1, 2, 3, 4$. Em seguida, esses grupos são usados para formar quatro metades, de modo que os modelos calculados com as metades 1 e 3 são validados com as metades 2 e 4, respectivamente, como mostra a Figura 3.3.

O Algoritmo 1 descreve como é feita a SHV usando o TCC. O algoritmo recebe como argumentos os modelos a serem comparados, m e n, e o número de componentes dos modelos, c. Nas linhas 4 e 5, a função TCC aplica a fórmula da Equação 2.7. O parâmetro $m_B(:, i)$ da função TCC na linha 5 representa o componente i do modo B do modelo m. Os parâmetros $n_B(:, j)$, $m_C(:, i)$ e $n_C(:, j)$ são interpretados da mesma forma.

Algoritmo 1 Validação usando o TCC

```
1: procedure VALIDTCC(m, n, c)
2:     for i ← 1, c do
3:         for j ← 1, c do
4:             btcc(i, j) ← TCC(m_B(:, i), n_B(:, j)))
5:             ctcc(i, j) ← TCC(m_C(:, i), n_C(:, j)))
6:         end for
7:     end for
8:     for i ← 1, c do
9:         for j ← 1, c do
10:            if btcc(i, j) > 0.95 and ctcc > 0.95 then
11:                match(i, j) ← 1
12:            else
13:                match(i, j) ← 0
14:            end if
15:        end for
16:    end for
17:    valid ← True
18:    for j ← 1, c do
19:        soma ← 0
20:        for i ← 1, c do
21:            soma ← soma + match(i, j)
22:        end for
23:        if soma ≠ 1 then
24:            valid ← False
25:        end if
26:    end for
27:    if valid then
28:        print Modelo validado
29:    else
30:        print Modelo não validado
31:    end if
32: end procedure
```

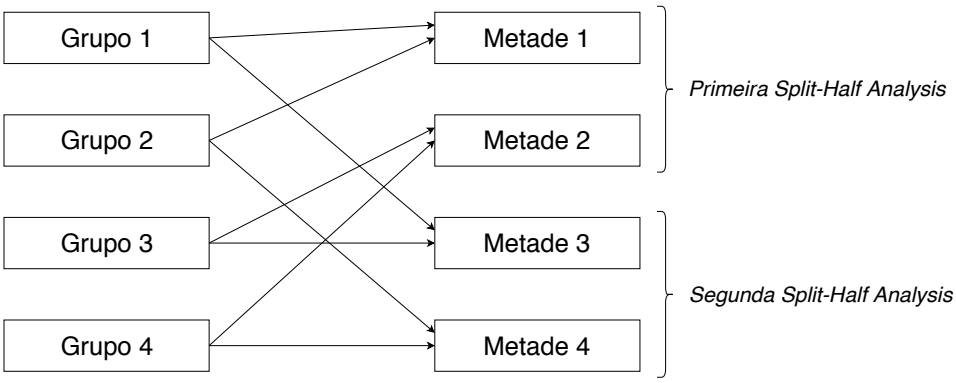


Figura 3.3: Divisão das amostras para realizar a *Split-Half Validation*

3.3 Clusterização

Neste trabalho será usada a Clusterização Hierárquica Aglomerativa, explicada na Seção 2.3, para agrupar os usuários usando o modelo PARAFAC obtido. Cada usuário será um elemento com coordenadas iguais aos *loadings* correspondentes no modo usuário. Antes de executar o algoritmo de clusterização, é feita uma normalização nos dados para evitar que se chegue a resultados falsos. Seja x uma coluna da matriz de dados e s a sua versão normalizada, a normalização é dada por

$$s(i) = \frac{x(i) - média(x)}{dp(x)} \tag{3.1}$$

onde $dp(x)$ é o desvio padrão de x.

Capítulo 4

Resultados

4.1 Análise de Fatores

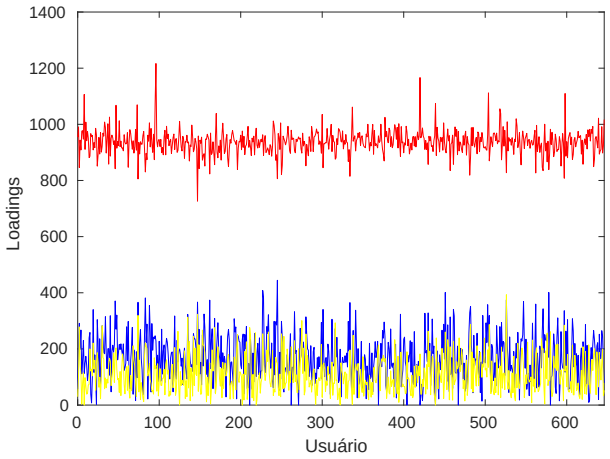
O primeiro passo na análise de fatores foi tentar validar modelos de 2 a 6 componentes. Para o tensor dos tráfegos, foram validados modelos com 2, 3 e 4 componentes. Para os tensores da latência e da perda, foram validados apenas modelos com 2 componentes. Após isso, foram calculados modelos com os números de componentes validados. Para avaliar a qualidade dos modelos, foram calculados os valores de variação explicada e de CORCONDIA. A Tabela 4.1 mostra os resultados obtidos.

Tabela 4.1: Valores de variação explicada e CORCONDIA para os modelos validados

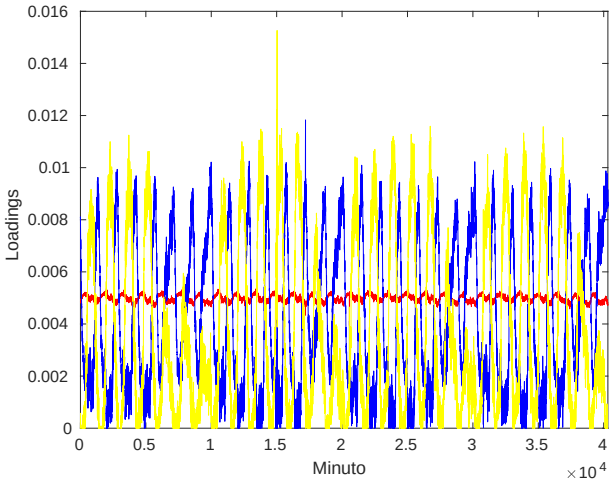
Tensor	Número de componentes	Variação explicada	CORCONDIA
Tráfegos	2	97,0	98,1
Tráfegos	3	97,2	99,6
Tráfegos	4	97,2	99,6
Latência	2	97,1	62,4
Perda	2	23,5	99,7

Todos os valores de variação explicada e de CORCONDIA foram altos para os modelos dos tráfegos e foi selecionado o modelo com 3 componentes para seguir com a análise. O modelo da latência com 2 componentes, embora tenha um valor baixo de CORCONDIA, tem um alto valor de variação explicada, então foi avaliado como um bom modelo. O modelo da perda com 2 componentes também foi considerado bom, no entanto, ao contrário do modelo da latência, tem alto valor de CORCONDIA e baixo valor de variação explicada. A Figura 4.1 mostra os *loadings* de cada modo do modelo dos tráfegos com 3 componentes. Na Figura 4.1(c), o 0 representa o tráfego de download e o 1 representa o tráfego de upload. É possível ver que no modo minuto há um padrão que se repete. Para observá-lo com mais clareza, foi feito um gráfico desse modo só com os minutos da primeira semana, que pode ser

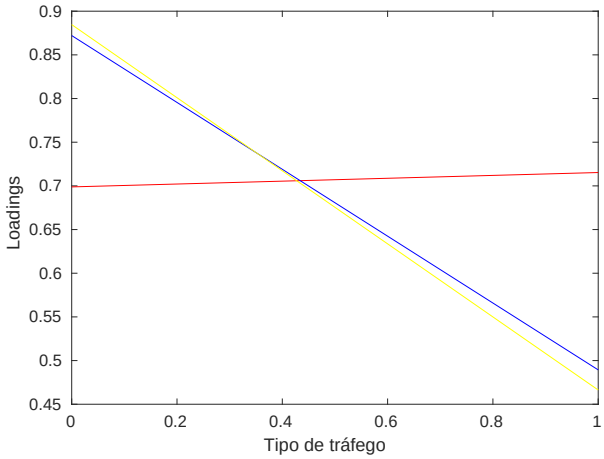
visto na Figura 4.2. Os dias estão separados por linhas verticais.



(a) Modo usuário



(b) Modo minuto



(c) Modo tipo de tráfego

Figura 4.1: Modelo do tensor dos tráfegos de download e upload

Na Figura 4.2, pode ser visto que há um padrão que se repete nos quatro primeiros dias e outro que se repete no três últimos. Também pode ser visto, na Figura 4.1(b), que, após a primeira semana, o primeiro padrão se repete cinco vezes e então o segundo se repete duas, depois novamente o primeiro se repete cinco vezes e o gráfico segue assim até o final. Como os 28 dias de dados foram escolhidos de modo que fossem quatro semanas começando na segunda-feira, pode-se dizer que o primeiro padrão está relacionado aos dias úteis e o segundo aos finais de semana e feriados.

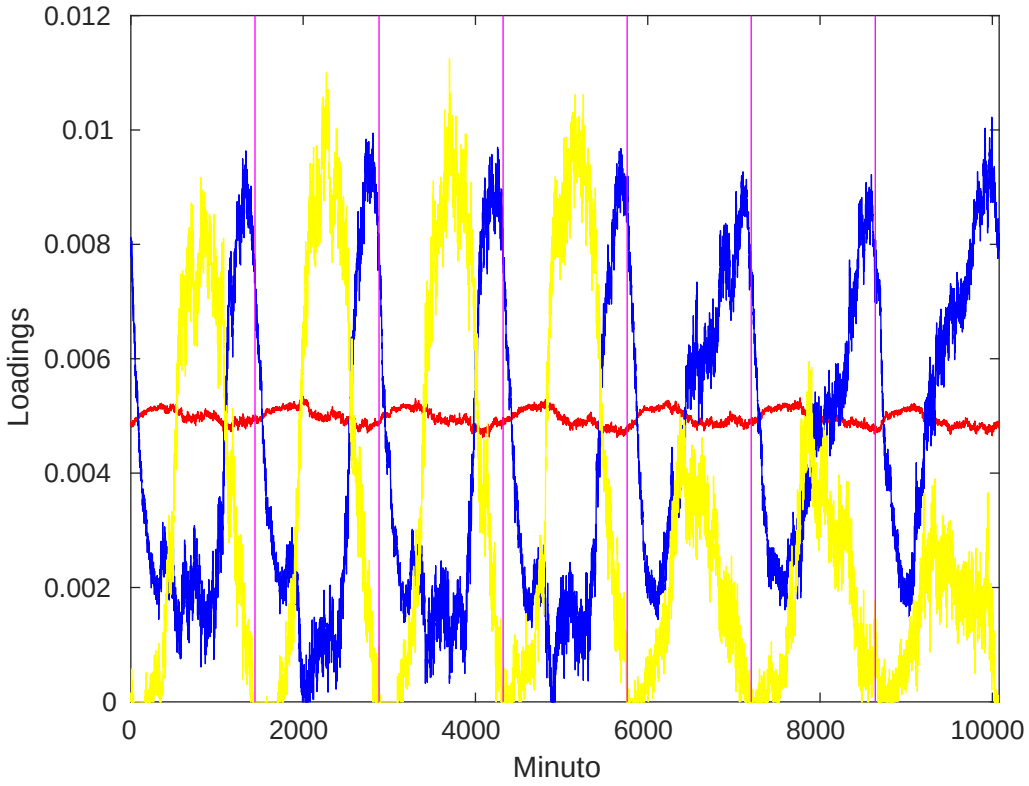


Figura 4.2: Primeira semana do gráfico da Figura 4.1(b)

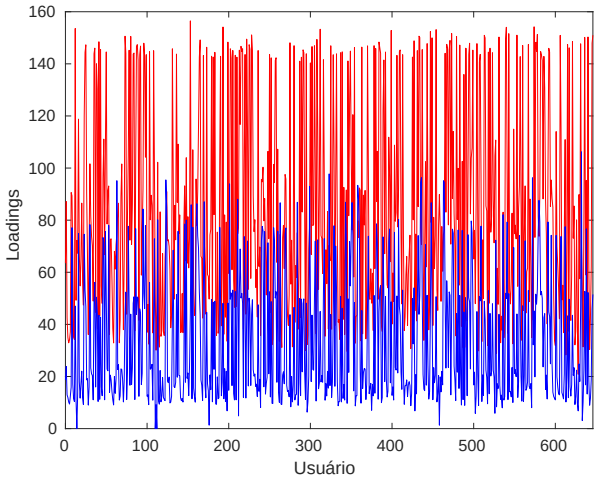
A primeira semana apresenta um comportamento diferente, com o segundo padrão nos últimos três dias. Isso pode ser explicado pelo fato de a sexta-feira da primeira semana ser o dia 7 de setembro, o Dia da Independência do Brasil, em que os usuários apresentaram um comportamento típico de final de semana por ser um feriado.

Outra característica interessante que pode ser observada na Figura 4.2 é que, nos dias úteis, o componente amarelo está associado ao período diurno e o componente azul está associado ao período noturno. Nos finais de semana, no entanto, o componente azul quase sempre é maior do que o componente amarelo. O componente vermelho, diferente dois outro dois, apresenta um comportamento quase linear, não mudando nem nos finais de semana.

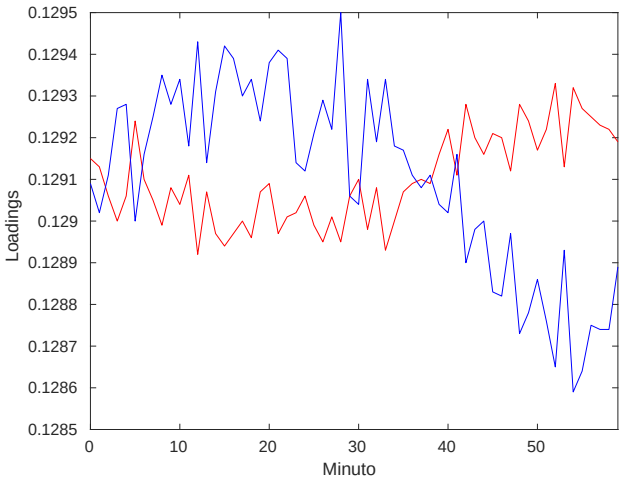
A Figura 4.3 apresenta os *loadings* encontrados para o modelo da latência. Como

pode ser visto, os modos minuto e hora não apresentam nenhum padrão visual claro. Isso faz sentido para o modo minuto, pois não se espera que todos os minutos da hora tenham comportamento parecido. Para o modo hora, no entanto, faria sentido que houvesse um padrão diário e semanal, pois se espera que a cada dia se repita um comportamento de acordo com a hora e que a cada semana se repita um comportamento de acordo com o dia.

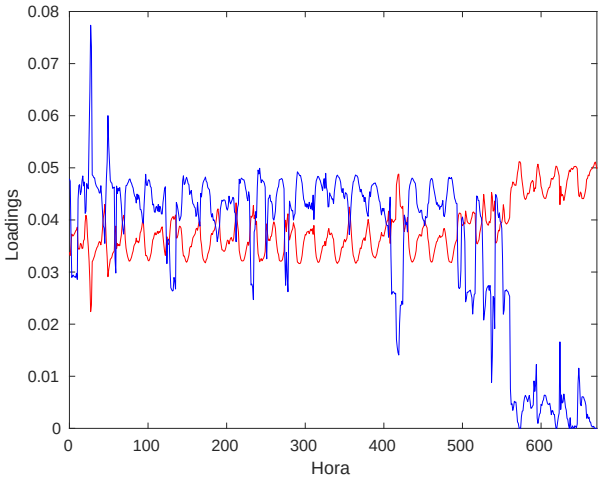
A Figura 4.4 apresenta os *loadings* encontrados para o modelo da perda. Assim como no modelo da latência, não há nenhum padrão visual no modo minuto, o que, como já foi dito, faz sentido. O modo hora, entretanto, desconsiderando os picos encontrados no início e no fim do gráfico, apresenta um padrão diário, embora ele não seja tão claro quanto os padrões encontrados no modelo dos tráfegos. Uma diferença no padrão diário do modelo da perda é que ele parece não ser influenciado pelo dia da semana.



(a) Modo usuário

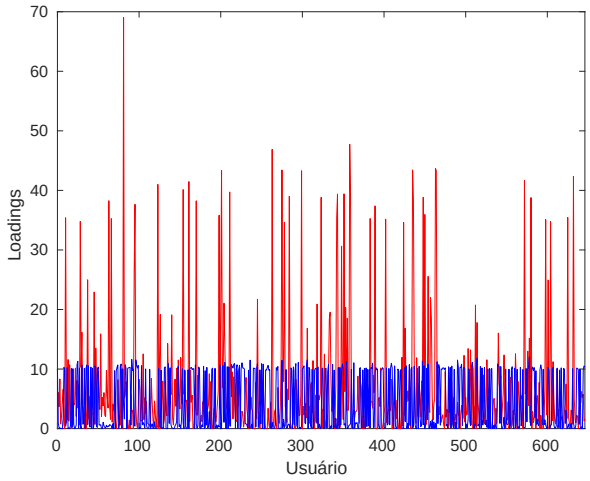


(b) Modo minuto

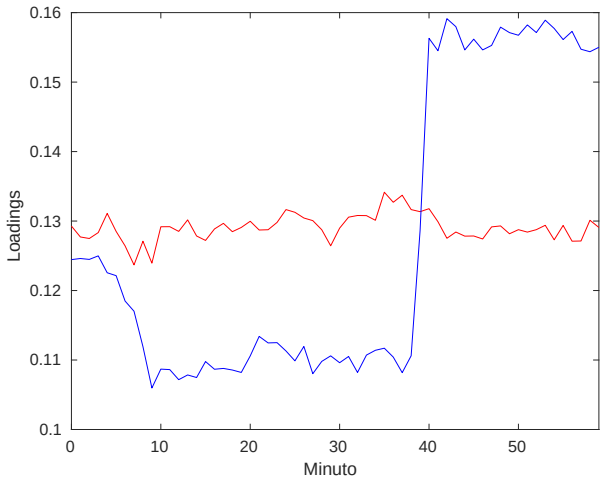


(c) Modo hora

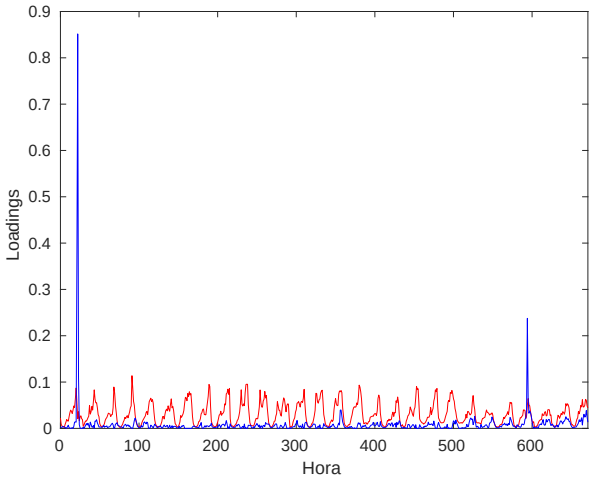
Figura 4.3: Modelo do tensor da latência



(a) Modo usuário



(b) Modo minuto



(c) Modo hora

Figura 4.4: Modelo do tensor da perda

4.2 Clusterização

Como foi visto na Seção 3.3, os *loadings* encontrados usando o PARAFAC foram usados para clusterizar os usuários usando a CHA. No caso do modelo dos tráfegos, foram considerados apenas os componentes azul e amarelo, uma vez que o componente vermelho não varia significativamente em nenhum dos modos. Assim, para os três modelos, os usuários possuem duas coordenadas. Foram feitas três clusterizações independentes, cada uma usando um dos modelos. A Tabela 4.2 mostra o tamanho dos clusters para os três modelos.

Tabela 4.2: Tamanho dos clusters encontrados

Modelo / Cluster	Tráfegos	Latência	Perda
1	458	391	257
2	157	102	349
3	29	154	37
4	2	–	3
5	1	–	1
Soma	647	647	647

4.2.1 Clusterização usando o modelo dos tráfegos

A Figura 4.5 mostra o resultado da clusterização usando o modelo dos tráfegos. Decidiu-se cortar o dendrograma na altura 2, dando origem a 5 clusters. Como pode-se observar na Tabela 4.2, um usuário ficou isolado em um cluster, o que indica que ele é muito diferente do restante, por isso decidiu-se desconsiderá-lo.

Com os clusters definidos, foram feitos gráficos das médias e medianas por minuto de cada cluster para cada métrica. Para apresentar nos gráficos, foi escolhida, para cada métrica, a semana que melhor caracterizava os dados e mostrava com mais clareza a diferença de comportamento entre os clusters. Para os clusters do modelo dos tráfegos, foi escolhida a semana 3 para montar os gráficos. A Figura 4.6 mostra as médias do tráfego de download dos clusters encontrados usando os tráfegos. Como pode-se observar, os clusters 1 e 2 apresentam um padrão diário claro, com poucas diferenças entre eles. Embora esses dois clusters apresentem padrões visuais muito parecidos, há uma grande diferença nas quantidades médias de tráfego gerado por eles. Os usuários do cluster 2 geram muito mais tráfego, tendo picos diários entre 1,4 e 2,3 megabytes, já os usuários do cluster 2 geram picos diários entre 0,8 e 0,5 megabytes. No cluster 3, embora seja possível observar um padrão diário, ele não está tão bem definido quanto nos dois primeiros clusters. Pode-se notar que as médias desse cluster são bem próximas das médias do cluster 2, com exceção do final

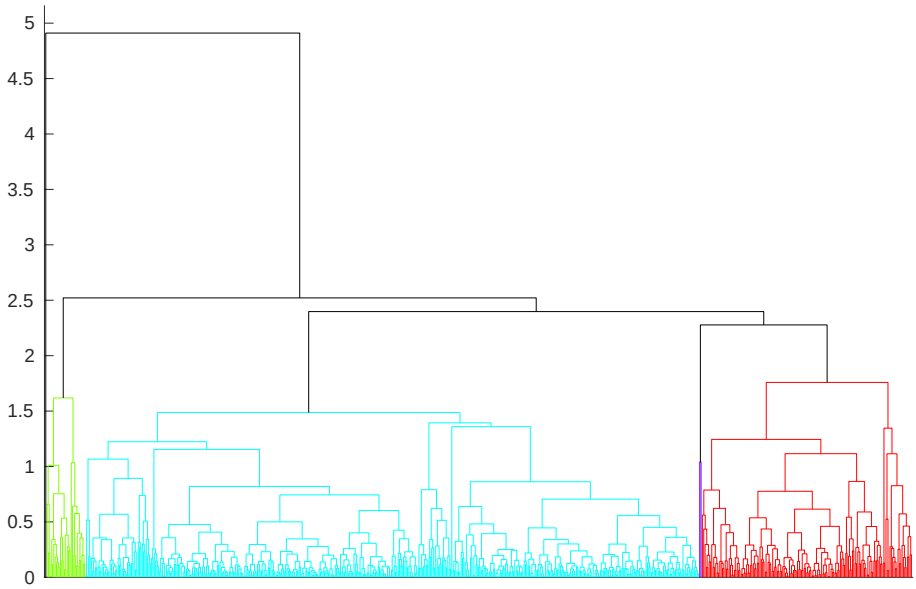
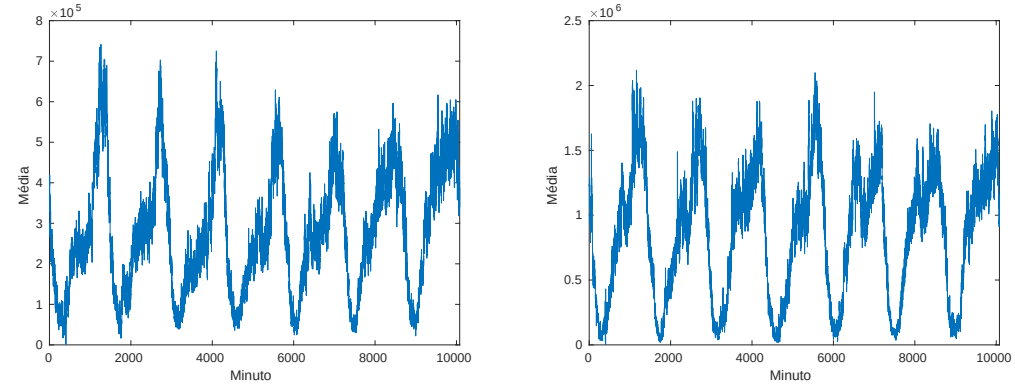


Figura 4.5: Dendrograma da clusterização pelo modelo dos tráfegos. As cores identificam os clusters formados cortando o dendrograma na altura 2

de semana, quando ocorre uma considerável redução na quantidade de tráfego. O cluster 4, por sua vez, não apresenta nenhum padrão visual e, além disso, apresenta médias muito mais altas do que os outros clusters, com picos diários entre 8 e 15 MB. Outra característica interessante do cluster 4 é que as médias na sexta-feira, sábado e domingo são consideravelmente mais altas do que nos outros dias, mostrando que esses usuários geram mais tráfego no final de semana.

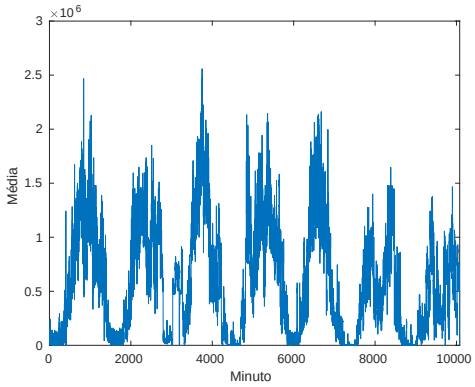
A Figura 4.7 mostra as medianas do tráfego de download dos clusters encontrados usando os tráfegos. Nela, a diferença entre os clusters 1 e 2 fica muito mais clara. Dos quatro clusters, o cluster 1 é o que apresenta as menores medianas e praticamente não tem picos diários no final de semana. O cluster 2 apresenta o mesmo comportamento das médias nas medianas. As medianas do cluster 3 apresentam o mesmo comportamento de suas médias, tendo um certo padrão nos dias úteis e um valor baixo no final de semana. O gráficos das medianas do cluster 4 é igual ao gráfico das médias, pois esse cluster tem apenas 2 usuários.

A Figura 4.8 mostras as médias do tráfego de upload dos clusters encontrados usando os tráfegos. É possível ver que os padrões são parecidos com aqueles encontrados nas médias do tráfego de download. Os padrões diários dos clusters 1 e 2 são bastante parecidos, mas o tráfego de upload gerado pelo cluster 2 é bem maior. O cluster 3 novamente apresenta um padrão não muito claro e médias bem superiores às dos dois primeiros clusters. Diferente do tráfego de download, o tráfego de upload sofre uma redução já no quarto dia da semana. O cluster 4 novamente não apresenta

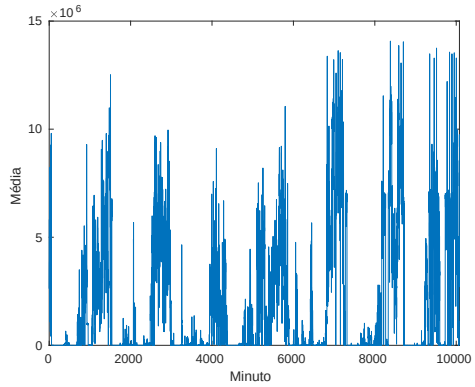


(a) Cluster 1 (458 usuários)

(b) Cluster 2 (157 usuários)



(c) Cluster 3 (29 usuários)



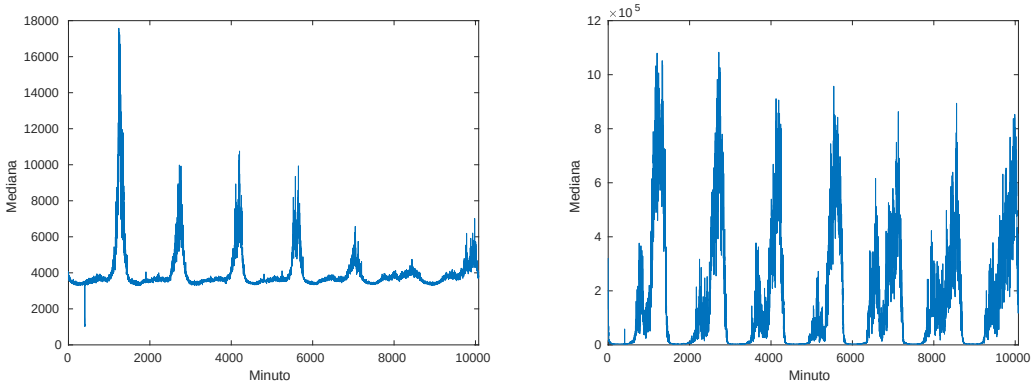
(d) Cluster 4 (2 usuários)

Figura 4.6: Média do tráfego de download por minuto de cada cluster na semana 3

nenhum padrão e gera muito mais tráfego do que os demais clusters.

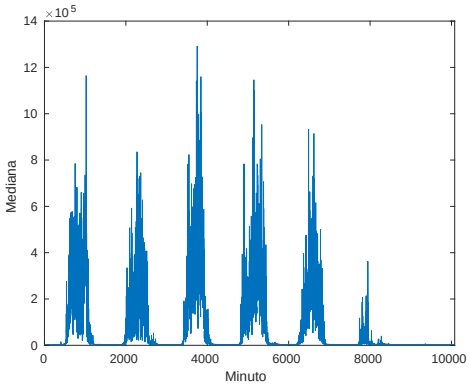
A Figura 4.9 mostra as medianas do tráfego de upload dos clusters. O cluster 1 apresenta um padrão de tráfego de upload muito parecido o do tráfego de download, com a diferença que os dias úteis são mais uniformes. O cluster 2 apresenta um padrão diário muito mais claro do que na mediana do tráfego de download e um valor superior nas medianas do que o cluster 1. Com o que foi visto até aqui, pode-se dizer que os clusters 1 e 2 possuem usuários que tem um comportamento diário bem definido, sendo que os usuários do cluster 2 geram mais tráfego, tanto de download quanto de upload. O comportamento das medianas do cluster 3 para o tráfego de upload é igual ao do tráfego de download. Assim, podemos dizer que os usuários desse cluster geram bastante tráfego nos dias úteis, mas diminuem consideravelmente nos finais de semana. Com relação ao cluster 4, pode-se dizer que são dois usuários que geram uma quantidade de tráfego muito acima do normal e não possuem um padrão diário.

A Figura 4.10 mostra a função distribuição acumulada complementar (CCDF) do tráfego de download dos clusters. Pode-se ver que a CCDF dos clusters 1, 2 e 3 é similar, com exceção da cauda. A CCDF do cluster 2 possui cauda mais longa que

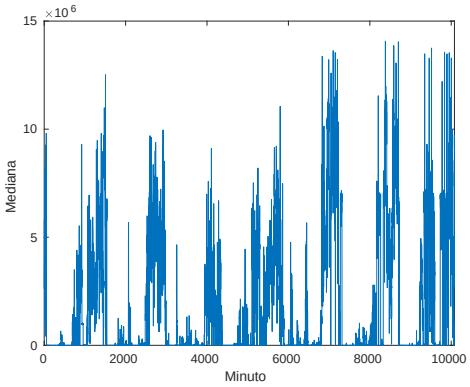


(a) Cluster 1 (458 usuários)

(b) Cluster 2 (157 usuários)



(c) Cluster 3 (29 usuários)

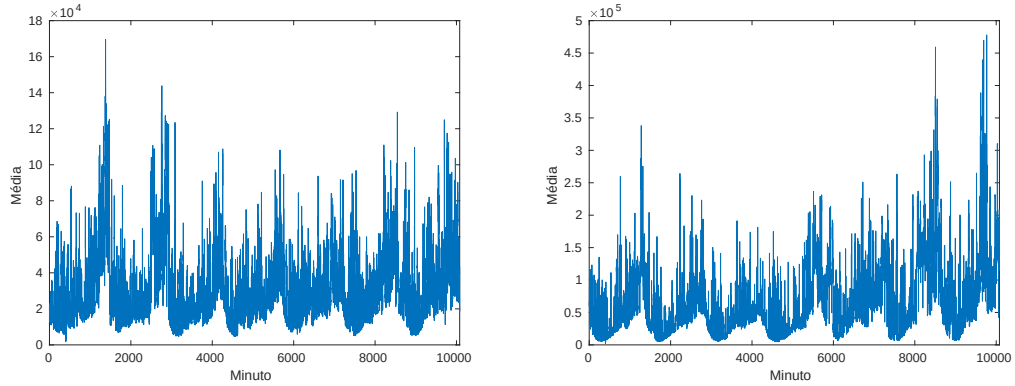


(d) Cluster 4 (2 usuários)

Figura 4.7: Mediana do tráfego de download por minuto de cada cluster na semana 3

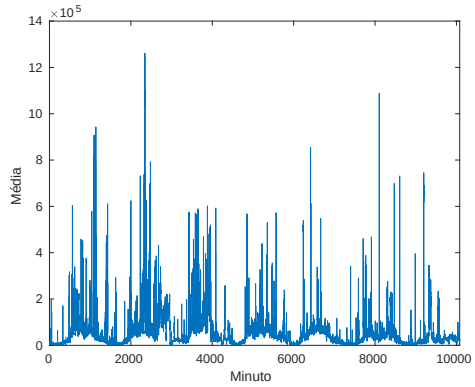
as demais. Para o cluster 2, a probabilidade do tráfego de download medido em 1 minuto ser maior que 90 MB é aproximadamente 10^{-5}, já para os clusters 1 e 3 essa probabilidade é zero. O cluster 4 se diferencia dos demais, pois o tráfego de download por minuto medido nos usuários desse cluster é sempre inferior a 20 MB. Nos demais clusters, o valor do tráfego pode atingir valores bem maiores. É interessante notar que, embora o cluster 4 tenha médias e medianas superiores àquelas dos demais clusters, os maiores picos de tráfego em um minuto são registrados nesses clusters, o que indica que esses picos são valores muito distantes do tráfego médio dos usuários desses clusters. Vale lembrar que o cluster 4 só possui dois usuários, representando um caso particular.

A Figura 4.11 mostra o boxplot do tráfego de download dos clusters. As medianas dos clusters 1, 3 e 4 tem valores muito próximos e também são bem próximas do primeiro quartil. As diferenças entre esses clusters aparece no terceiro quartil. O tráfego de download dos usuários do cluster 1 possui menor variabilidade que o dos clusters 3 e 4, 75% do tráfego de download dos usuários do cluster 1 é inferior a 10 KB, já para os clusters 3 e 4, 75% do tráfego de download é inferior a 100 KB e 1

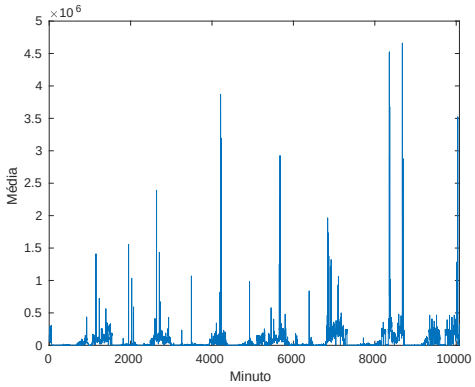


(a) Cluster 1 (458 usuários)

(b) Cluster 2 (157 usuários)



(c) Cluster 3 (29 usuários)



(d) Cluster 4 (2 usuários)

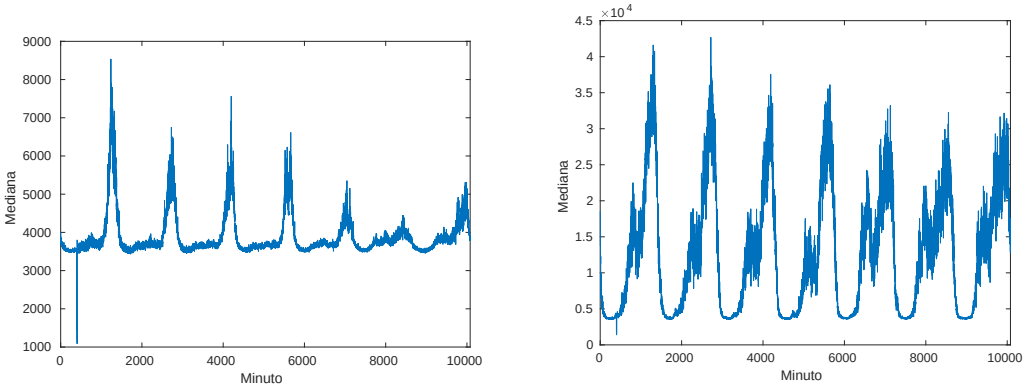
Figura 4.8: Média do tráfego de upload por minuto de cada cluster na semana 3

MB, respectivamente. A mediana do tráfego de download dos usuários do cluster 2, por sua vez, é aproximadamente uma ordem de grandeza superior à mediana dos outros clusters.

A Figura 4.12 mostra a CCDF do tráfego de upload dos clusters. A CCDF do tráfego de upload mostra que o cluster 1 é, de longe, o que apresenta os maiores valores para este tráfego, podendo atingir valores superiores a 70 MB. Os clusters 2 e 3 apresentam uma CCDF parecida, com valores de tráfego de upload inferiores aos do cluster 1 e não ultrapassando 40 MB. O cluster 4 manteve o mesmo comportamento do tráfego de download, apresentando valores bem menores para o tráfego de upload quando comparado com o tráfego dos outros clusters.

A Figura 4.13 mostra o boxplot do tráfego de upload dos clusters. Podemos notar que a variabilidade do tráfego de upload é menor que a do tráfego de download. Quanto à mediana, ela possui valores similares para os quatro clusters e também possui valores próximos à mediana do tráfego de download.

Com o objetivo de analisar as diferenças nos tráfegos gerados durante a madrugada e no restante do dia, foram gerados boxplots para esses períodos específicos. As figuras 4.14 e 4.15 mostram, respectivamente, os tráfegos de download coletados



(a) Cluster 1 (458 usuários)

(b) Cluster 2 (157 usuários)

(c) Cluster 3 (29 usuários)

(d) Cluster 4 (2 usuários)

Figura 4.9: Mediana do tráfego de upload por minuto de cada cluster na semana 3

entre 1h e 7h e entre 7h e 1h para cada cluster. Pode-se observar que grande parte do tráfego gerado por minuto durante a madrugada fica restrito a um pequeno intervalo, uma vez que os limites superior e inferior das caixas estão bastante próximos, mostrando pouca variabilidade para o tráfego neste período. A única exceção é o cluster 2, que possui limites mais afastados. Outro cluster que se destaca nessa faixa de horário é o cluster 1, que é o único que possui *outliers* abaixo de 100 bytes, mostrando que em certos momentos praticamento nenhum tráfego foi gerado por um ou mais usuários desse cluster. Com relação ao horário de 7h até 1h, ou seja, manhã, tarde e noite, o comportamento apresentado é bastante parecido com o do boxplot que considera as 24h do dia, o que mostra que o tráfego no período da madrugada, apesar de apresentar estatísticas bem diferentes, não tem grande influência nas estatísticas obtidas para o período de 24h.

As figuras 4.16 e 4.17 mostram, respectivamente, os boxplots do tráfego de upload dos clusters para os períodos de 1h a 7h e de 7h a 1h. Pode-se observar que o resultado é muito parecido com o obtido para o tráfego de download.

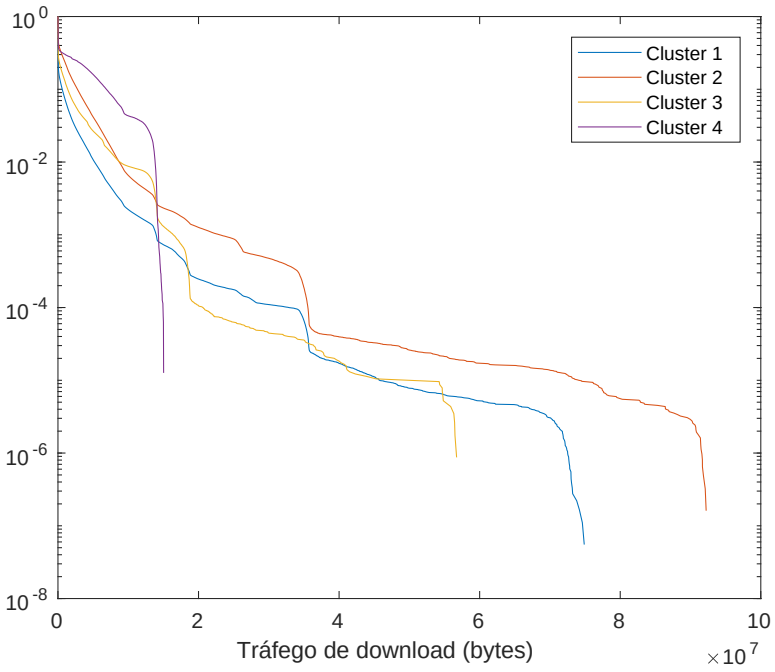


Figura 4.10: CCDF dos clusters do tráfego de download

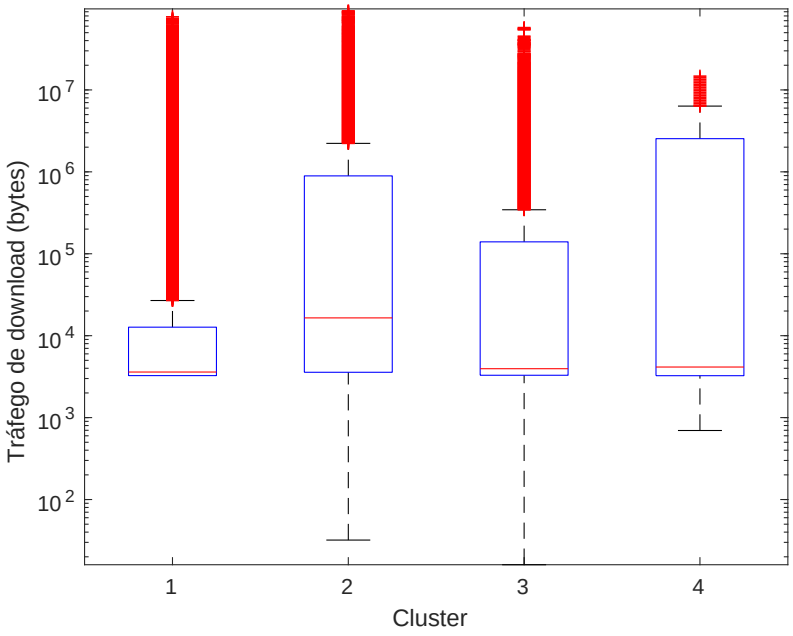


Figura 4.11: Boxplot dos clusters do tráfego de download

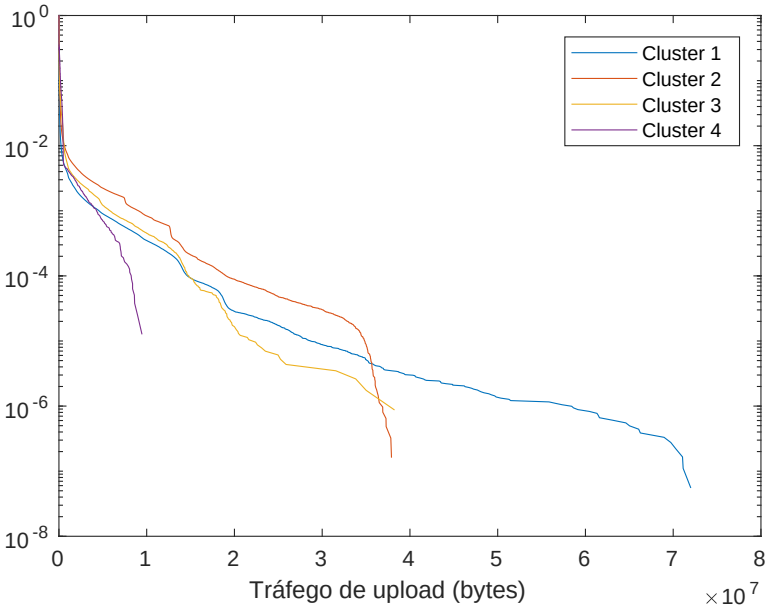


Figura 4.12: CCDF dos clusters do tráfego de upload

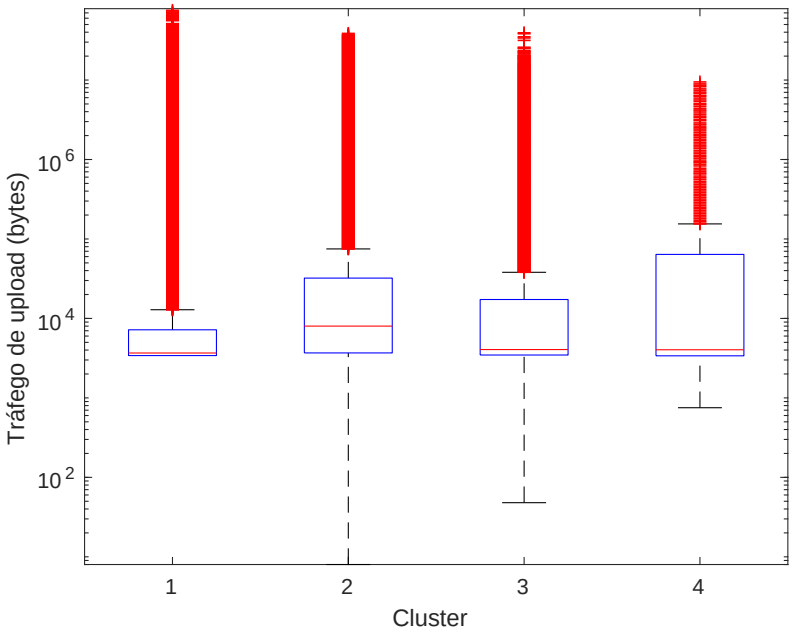


Figura 4.13: Boxplot dos clusters do tráfego de upload

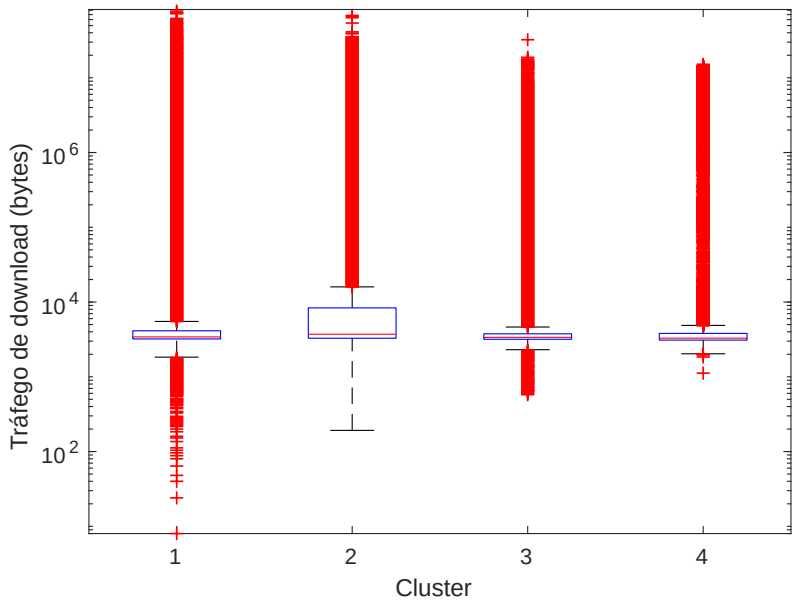


Figura 4.14: Boxplot dos clusters do tráfego de download no período de 1h até 7h

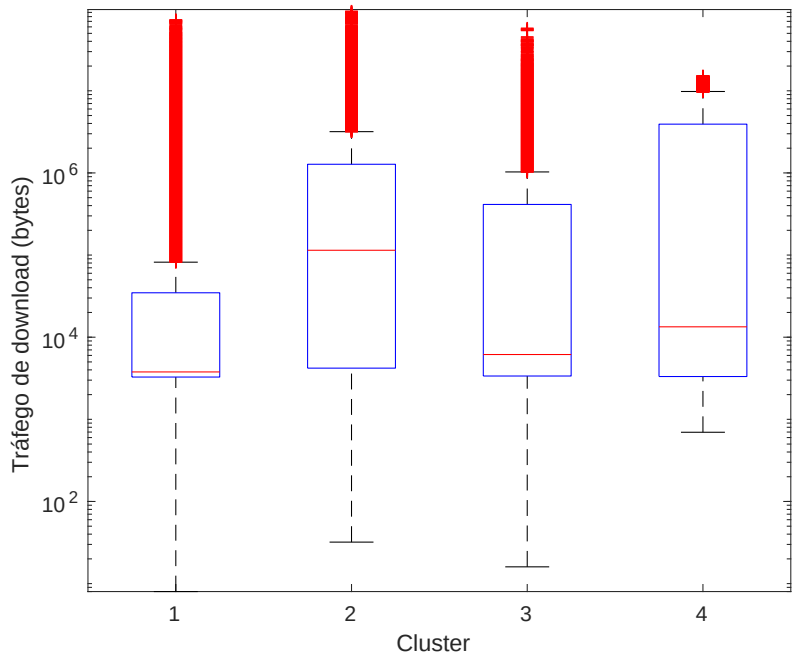


Figura 4.15: Boxplot dos clusters do tráfego de download no período de 7h até 1h

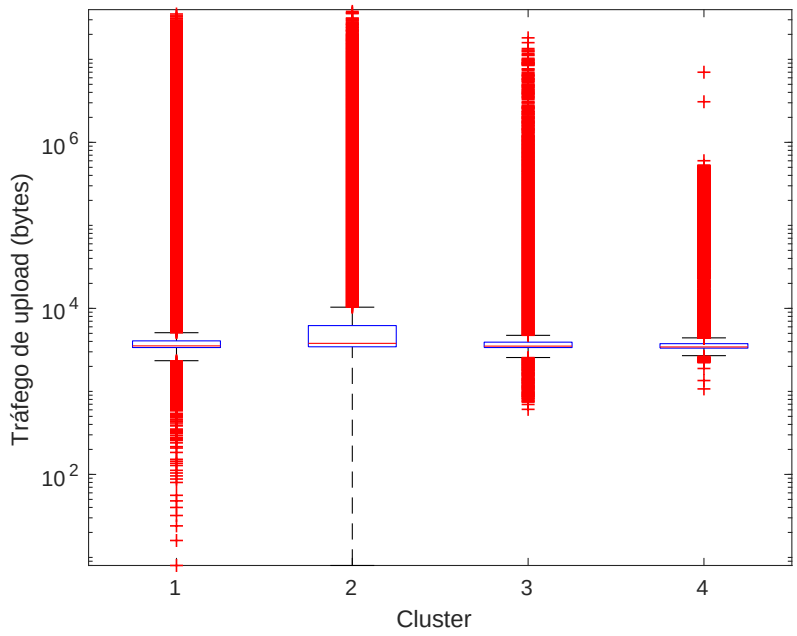


Figura 4.16: Boxplot dos clusters do tráfego de upload no período de 1h até 7h

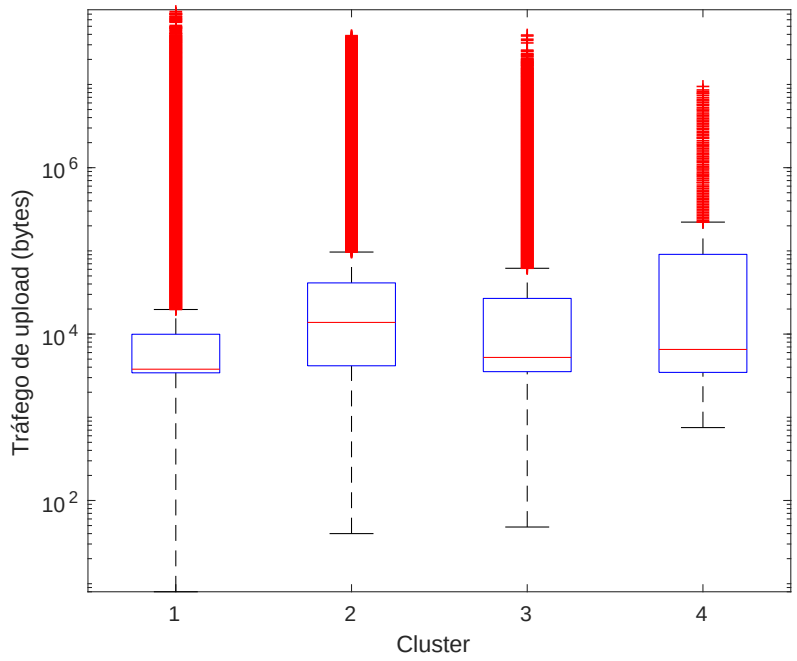


Figura 4.17: Boxplot dos clusters do tráfego de upload no período de 7h até 1h

4.2.2 Clusterização usando o modelo da latência

A Figura 4.18 mostra o resultado da clusterização usando o modelo da latência. Decidiu-se cortar o dendrograma na altura 1, dando origem a 3 clusters.

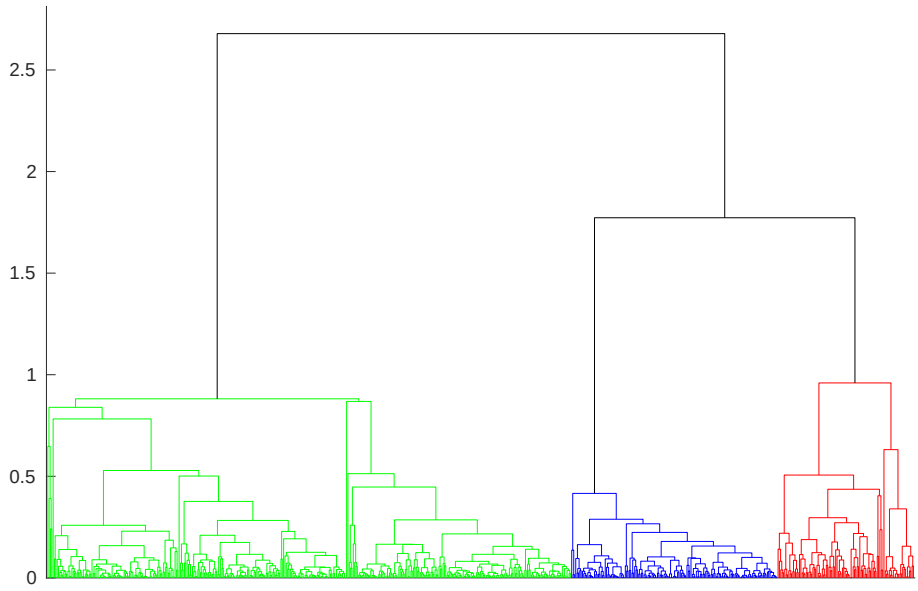
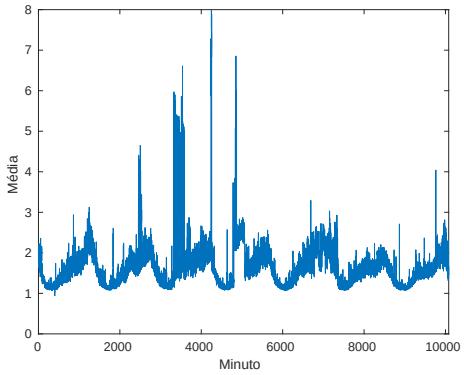


Figura 4.18: Dendrograma da clusterização pelo modelo da latência. As cores identificam os clusters formados cortando o dendrograma na altura 1

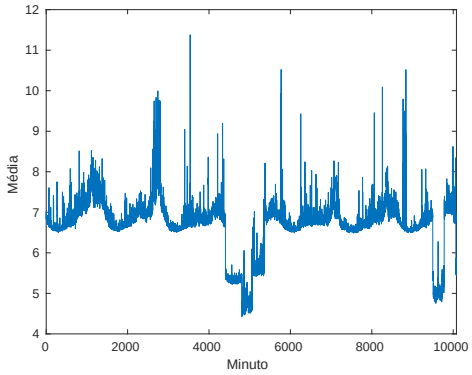
Para os clusters do modelo da latência, também foi escolhida a semana 3 para montar os gráficos da média e mediana do tráfego durante a semana. A Figura 4.19 mostra as médias da latência dos clusters. Com relação ao padrão diário, os clusters 1 e 3 apresentam um padrão: latência com valores maiores no final do dia. Os clusters 2 e 3 apresentam as maiores médias de latência.

A Figura 4.20 mostra as medianas da latência dos clusters. Como pode ser visto, o cluster 1 é o que apresenta o padrão mais claro: latências maiores no final do dia. Esse cluster é novamente o que apresenta os valores de latência mais baixos. O cluster 2 não apresenta um padrão e possui medianas bem mais baixas em alguns horários da semana. O cluster 3 é o que apresenta as medianas mais altas. Pode-se concluir que o cluster 1 possui usuários que enfrentam os menores atrasos em suas redes e que esses atrasos seguem um padrão diário que não é influenciado pelo dia da semana. Com relação ao cluster 2, são usuários que possuem atrasos maiores que os do cluster 1 e cuja latência não possui um padrão diário claro. O cluster 3 possui usuários que também enfrentam atrasos maiores e seguem um padrão diário semelhante aos usuários do cluster 1.

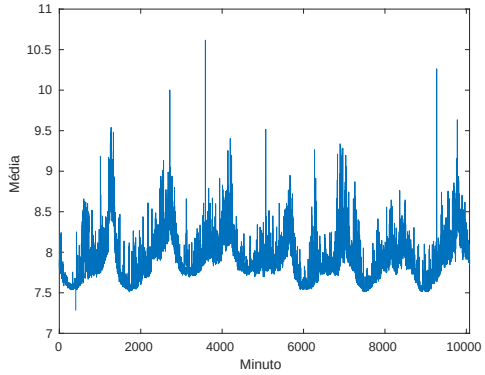
A Figura 4.21 mostra a CCDF da latência dos clusters. Até a probabilidade de



(a) Cluster 1 (391 usuários)



(b) Cluster 2 (102 usuários)

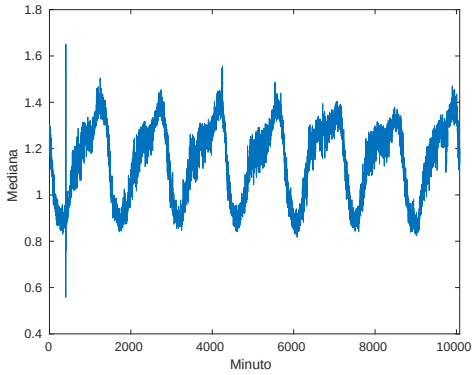


(c) Cluster 3 (154 usuários)

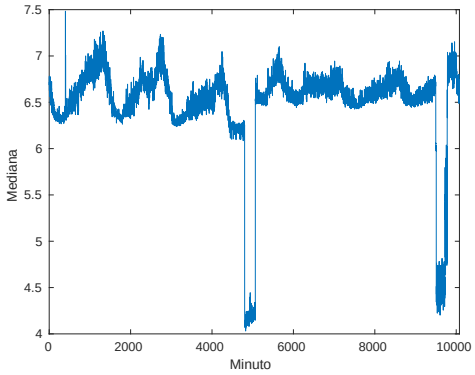
Figura 4.19: Média da latência por minuto de cada cluster na semana 3

10^{-4} todos os clusters apresentam uma distribuição parecida. O cluster 1 apresenta os valores mais altos para latência, seguido do cluster 2. A probabilidade dos usuários do cluster 3 experimentarem uma latência maior que 250 ms é praticamente zero.

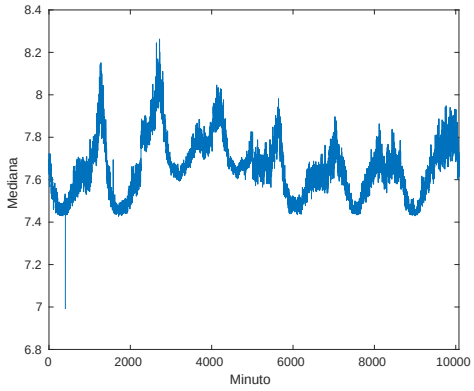
A Figura 4.22 mostra o boxplot para a latência dos clusters. Novamente o cluster 1 se destaca pelo baixo valor de mediana quando comparado com os demais, a mediana da latência é levemente superior a 1 ms. Os demais clusters tem medianas próximas de 10 ms e, além disso, os seus primeiros e terceiros quartis estão muito próximos, o que mostra que grande parte dos valores da latência está concentrada próxima da mediana.



(a) Cluster 1 (391 usuários)



(b) Cluster 2 (102 usuários)



(c) Cluster 3 (154 usuários)

Figura 4.20: Mediana da latência por minuto de cada cluster na semana 3

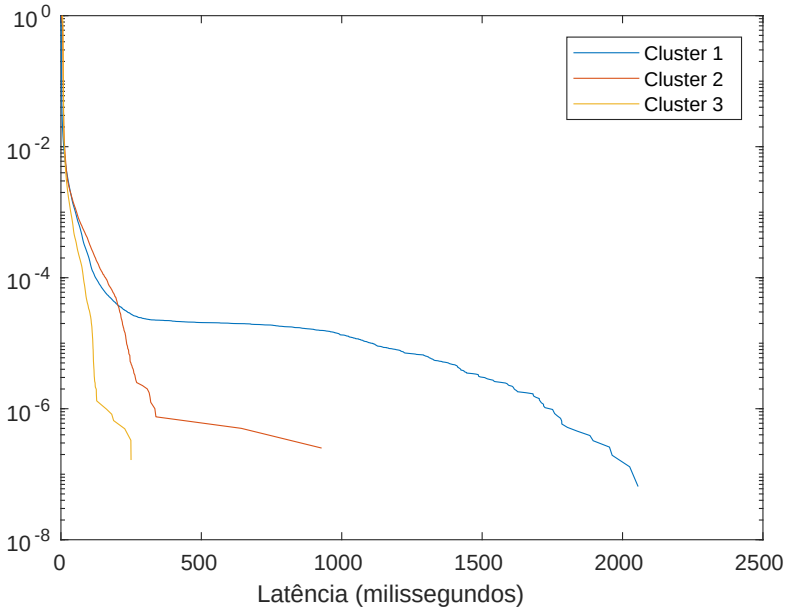


Figura 4.21: CCDF da latência dos clusters

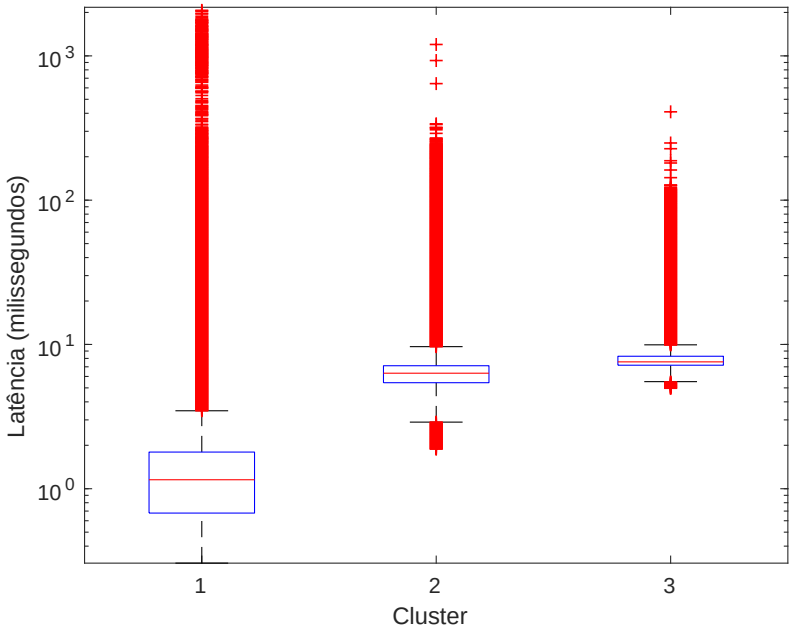


Figura 4.22: Boxplot dos clusters da latência

4.2.3 Clusterização usando o modelo da perda

A Figura 4.23 mostra o resultado da clusterização usando o modelo da perda. Decidiu-se cortar o dendrograma na altura 2, dando origem a 5 clusters. Essa clusterização também gerou um cluster com um único usuário e, assim como no caso do modelo dos tráfegos, decidiu-se desconsiderá-lo.

Para os clusters do modelo da perda, foram escolhidas diferentes semanas para cada cluster, com o objetivo de tentar identificar um padrão de perda para cada um deles. A Figura 4.24 mostra as médias da perda para os clusters. Os clusters 1, 2 e 3 apresentam um padrão diário claro: as perdas aumentam no final do dia. A média da perda é bastante parecida para os clusters 1 e 2. Já para o cluster 3, a média é bem superior, podendo chegar a 10 pacotes perdidos em uma rajada de 100. Com relação ao padrão semanal, os clusters 2 e 3 apresentam os maiores valores de perda de segunda a sexta, já o cluster 1, parece manter o mesmo padrão para todos os dias da semana. O cluster 4, por sua vez, não apresenta nenhum padrão diário nem semanal, e suas médias são muito superiores as médias dos outros clusters.

A Figura 4.25 mostra as medianas da perda para os clusters. Foram colocados apenas os gráficos dos clusters 3 e 4, pois as medianas dos clusters 1 e 2 se mantém em zero durante toda a semana. Assim como as médias, as medianas do cluster 4 são bem maiores do que as dos outros clusters.

Como foi visto na Figura 4.24 e na Figura 4.25, pode-se dizer que os usuários dos clusters 1 e 2 enfrentam uma perda de pacotes bem baixa, e possuem padrão diário e semanal. Os usuários dos clusters 3 e 4 enfrentam uma perda de pacotes

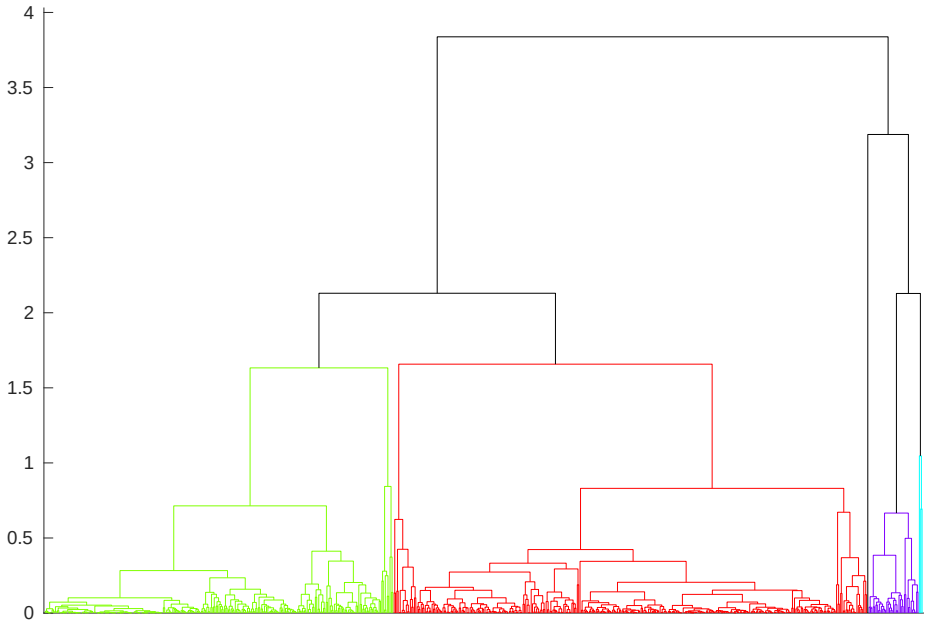
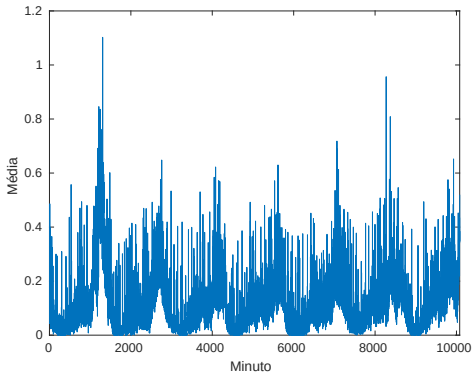


Figura 4.23: Dendrograma da clusterização pelo modelo da perda. As cores identificam os clusters formados cortando o dendrograma na altura 2

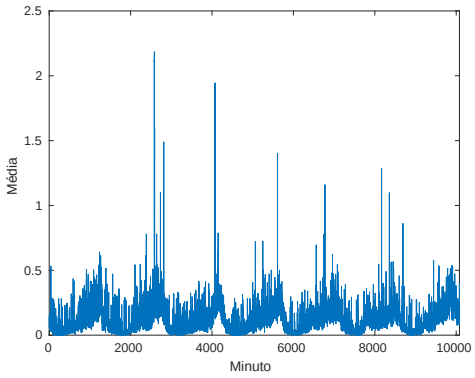
mais alta, com medianas acima de zero em boa parte da semana.

A Figura 4.26 mostra a CCDF da perda para os clusters. Nesse gráfico, foram considerados apenas os valores maiores que 0, para observar a distribuição condicional da perda. Pode-se observar que os clusters de 1 a 3 tem distribuições praticamente iguais. É interessante observar que os clusters 1 e 2, que, como foi visto, apresentam as menores médias e medianas de perda, podem ter valores muito altos de perda, a probabilidade da perda ser maior do que 90 ms é aproximadamente 10^{-4}. Com o cluster 4 acontece o oposto, pois ele é o que apresenta as maiores médias e medianas e, pela sua distribuição, pode ser visto que os maiores valores de perda registrados para esses usuários fica abaixo de 50 ms.

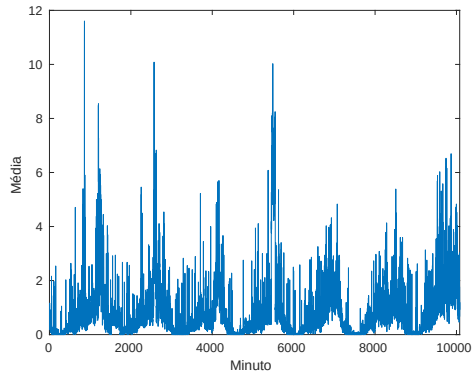
A Figura 4.27 mostra o boxplot da perda para os clusters. Neste gráfico, novamente não foram considerados os zeros. Os clusters 1 e 2 apresentam o mesmo comportamento no boxplot, com mediana 1 e terceiro quartil igual a 3. O cluster 3 também tem o terceiro quartil igual a 3, mas sua mediana é 2. O cluster 4 é o que possui menos *outliers*, mas também é que possui o maior terceiro quartil, que é igual a 8.



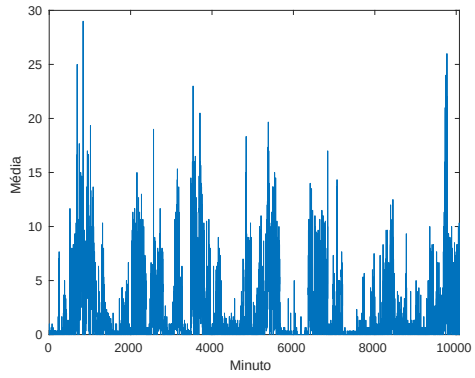
(a) Cluster 1 (257 usuários), semana 3



(b) Cluster 2 (349 usuários), semana 1

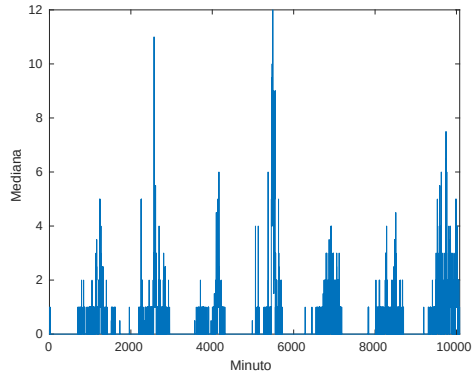


(c) Cluster 3 (37 usuários), semana 1

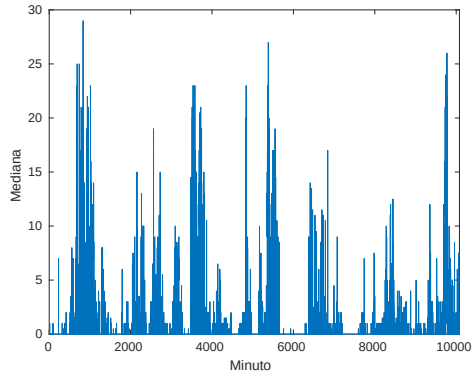


(d) Cluster 4 (3 usuários), semana 3

Figura 4.24: Média da perda por minuto de cada cluster



(a) Cluster 3 (37 usuários), semana 1



(b) Cluster 4 (3 usuários), semana 3

Figura 4.25: Mediana da perda por minuto de cada cluster

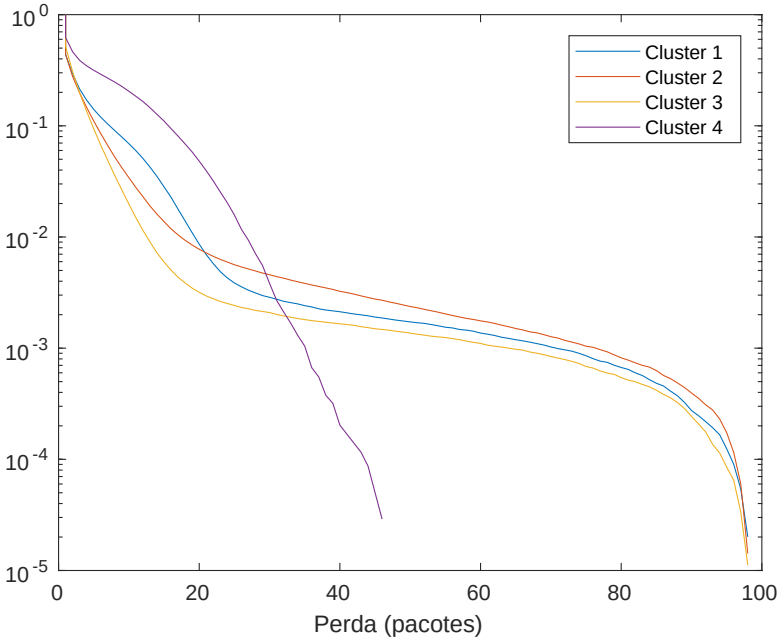


Figura 4.26: CCDF da perda dos clusters

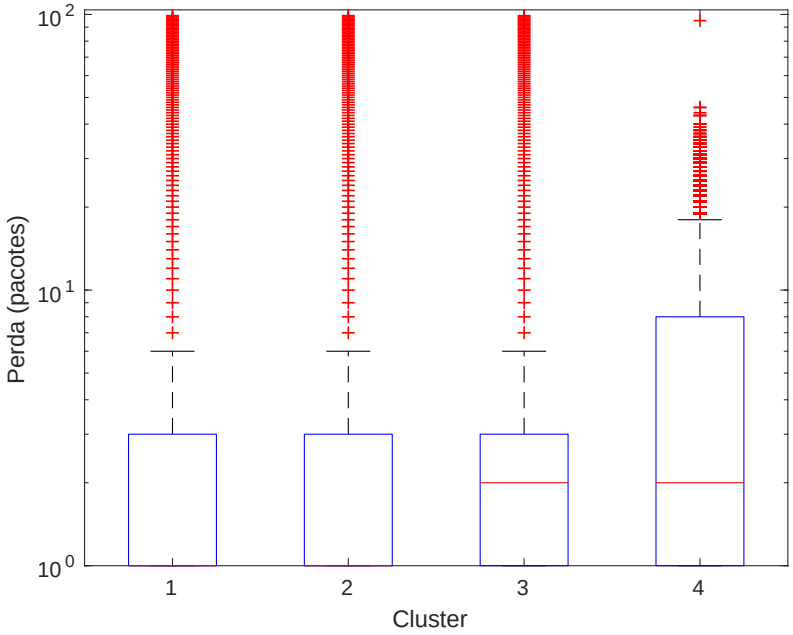


Figura 4.27: Boxplot da perda dos clusters

4.2.4 Interseção entre os clusters

A Tabela 4.3 mostra a interseção entre os clusters encontrados usando o modelo dos tráfegos, o modelo da latência e o modelo da perda. A primeira coluna mostra a quantidade de usuários na interseção e as demais trazem a identificação dos clusters de cada métrica de acordo com a Tabela 4.2. Os números entre parênteses mostram o tamanho de cada cluster.

Tabela 4.3: Interseção entre os clusters de cada métrica

Tamanho da interseção	Cluster dos tráfegos	Cluster da latência	Cluster da perda
235	1 (458)	1 (391)	2 (349)
106	1 (458)	3 (154)	1 (257)
85	2 (157)	1 (391)	2 (349)
48	1 (458)	2 (102)	1 (257)
39	2 (157)	3 (154)	1 (257)
27	1 (458)	1 (391)	1 (257)
17	1 (458)	2 (102)	3 (37)
16	2 (157)	2 (102)	1 (257)
15	3 (29)	1 (391)	2 (349)
13	1 (458)	1 (391)	3 (37)
12	2 (157)	1 (391)	1 (257)
8	1 (458)	2 (102)	2 (349)
4	3 (29)	2 (102)	3 (37)
4	3 (29)	3 (154)	1 (257)
3	2 (157)	2 (102)	3 (37)
2	1 (458)	3 (154)	2 (349)
2	2 (157)	2 (102)	2 (349)
2	3 (29)	2 (102)	1 (257)
1	1 (458)	1 (391)	5 (1)
1	1 (458)	2 (102)	4 (3)
1	3 (29)	1 (391)	1 (257)
1	3 (29)	2 (102)	4 (3)
1	3 (29)	3 (154)	2 (349)
1	3 (29)	3 (154)	4 (3)
1	4 (2)	1 (391)	1 (257)
1	4 (2)	3 (154)	1 (257)
1	5 (1)	1 (391)	2 (349)

Capítulo 5

Conclusão

Neste trabalho definimos perfis de usuários de um provedor médio da Internet a partir de dados de tráfego, latência e perda coletados em roteadores residenciais. Acreditamos que esses perfis podem ser muito úteis aos provedores para realizar tarefas de gerenciamento e planejamento de capacidade da rede.

O primeiro passo deste trabalho foi modelar os dados usando tensores e usar o método PARAFAC para reduzir a dimensionalidade e tentar encontrar algum padrão nos dados coletados. Com relação à diminuição da dimensionalidade, se fossem usados na clusterização os dados coletados, teríamos da ordem de 40.000 valores associados a cada usuário. Usando o PARAFAC, cada usuário foi representado por 2 valores. Conseguiu-se observar que os tráfegos de download e upload seguiam um padrão diário, que é diferente nos finais de semana e feriados, o que também dá origem a um padrão semanal. Foi visto também que a latência não seguiu um padrão claro ao longo dos 28 dias e que a perda seguia um certo padrão.

O segundo passo foi clusterizar os usuários usando os modelos PARAFAC obtidos. Foi possível ver que os clusters encontrados apresentavam diferenças significativas. Na clusterização usando o modelo dos tráfegos, foi visto que usuários de clusters diferentes, em geral, geram quantidades de tráfego diferentes. Também foi visto que alguns clusters apresentavam um padrão diário de tráfego, enquanto outros não, e que, para alguns clusters, o tráfego gerado variava de acordo com o dia da semana. Outro resultado interessante foi ver que o comportamento do tráfego gerado durante a madrugada é consideravelmente diferente daquele observado no restante do dia. Resultados parecidos foram encontrados para os tráfegos de download e upload.

Também foi possível encontrar clusters com padrões distintos para os modelos da latência e da perda. Foi visto que os três clusters definidos para a latência apresentavam latências médias significativamente diferentes ao longo da semana. O primeiro apresentava baixas latências médias, o segundo apresentava latências mais altas e o terceiro muito mais altas. Também foi visto que os maiores picos de latência de um usuário em um minuto foram encontrados nos clusters de menores latências

médias. Os clusters encontrados usando o modelo da perda apresentam grande diferença na quantidade de pacotes perdidos. Três dos quatros clusters apresentam um padrão diário e semanal bem definidos.

Os resultados obtidos mostram que há perfis bem definidos de usuários e entender como eles se comportam pode ser muito útil. Os ISPs podem usar esse tipo de estudo para ter um gerenciamento de rede muito mais eficiente. Os clusters que apresentam perdas ou latências altas, por exemplo, podem ser investigados pelo ISP para descobrir o que está acontecendo com esses usuários. Além disso, um modelo de comportamento de usuários normais pode ajudar a melhorar a segurança da rede, uma vez que pode ser usado para detectar tráfego anômalo.

Referências Bibliográficas

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

[2] 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*, pp. 1–12, ACM, 2018.

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

[4] KROONENBERG, P. M., *Applied multiway data analysis*, v. 702. John Wiley & Sons, 2008.

[5] HARSHMAN, R. A., "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory"multi-modal factor analysis", *UCLA Working Papers in Phonetics, 16*, pp. 1–84, 1970.

[6] JOLLIFFE, I., *Principal component analysis*. Springer, 2011.

[7] CARROLL, J. D., CHANG, J., "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young"decomposition", *Psychometrika*, v. 35, n. 3, pp. 283–319, 1970.

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

[9] ANDERSSON, C. A., BRO, R., "The N-way toolbox for MATLAB", *Chemometrics and intelligent laboratory systems*, v. 52, n. 1, pp. 1–4, 2000.

[10] 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.

[11] KRUSKAL, J. B., "Rank, decomposition, and uniqueness for 3-way and N-way arrays", *Multiway data analysis*, pp. 7–18, 1989.

[12] 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.

[13] BRO, R., KIERS, H. A., "A new efficient method for determining the number of components in PARAFAC models", *Journal of Chemometrics: A Journal of the Chemometrics Society*, v. 17, n. 5, pp. 274–286, 2003.

[14] 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.

[15] MÜLLNER, D., "fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python", *Journal of Statistical Software*, v. 53, n. 9, pp. 1–18, 2013.

[16] 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.

[17] 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.