Identificação de Anomalias em Redes de Dados baseada em Decomposição Tensorial

Ananda G. Streit, Gustavo H. A. Santos,
Rosa M.M. Leão, Edmundo de Souza e Silva, Daniel S. Menasché

Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brasil

{agstreit,gustavo,rosam,edmundo,sadoc}@land.ufrj.br

Abstract. *The problem of detecting anomalies in data networks has been widely studied and is a topic of fundamental importance. Many anomaly detection methods are based on packet inspection collected at the network core routers, with consequent disadvantages in terms of computational cost and privacy. We propose an alternative method in which package header inspection is not needed. The method is based on the extraction of a normal subspace obtained by the tensor decomposition technique considering the correlation between different metrics. Another advantage of our proposal is the interpretability of the obtained models. The flexibility of the proposal is illustrated by applying it to two distinct examples, both using actual data collected on residential routers.*

Resumo. *O problema de detectar anomalias em redes de dados tem sido amplamente estudado e é tópico de fundamental importância. Muitos métodos de detecção de anomalias fazem uso de inspeção de pacotes coletados no núcleo da rede, com consequentes desvantagens no custo computacional e privacidade. Propomos um método alternativo onde não é necessário inspecionar cabeçalhos de pacotes. O método é baseado na extração de um subespaço normal obtido pela técnica de decomposição de tensores considerando a correlação entre diferentes métricas. Outra vantagem é a interpretabilidade dos modelos obtidos. A flexibilidade da proposta é ilustrada aplicando-a em dois exemplos distintos, ambos usando dados reais coletados em roteadores residenciais.*

1. Introdução

O problema de detecção de eventos anômalos em uma rede de dados tem sido amplamente estudado devido à sua importância na determinação de eventos fora do padrão esperado, que impactam no funcionamento de uma rede, mas em geral são muito difíceis de identificar [Chandola et al. 2009]. O problema é desafiador pela grande variedade de anomalias, baixa frequência das ocorrências, e a determinação do que é o comportamento esperado. Um exemplo, dentre os inúmeros existentes, inclui mudanças pontuais nos padrões de tráfego em um canal de comunicação, causadas por um ataque de negação de serviço distribuído (DDoS), como o que ocorreu recentemente afetando serviços da Amazon [Fadilpašić 2019].

Em geral, a detecção de anomalias é baseada na análise dos cabeçalhos dos pacotes no núcleo da rede, com potencial elevado custo computacional, possíveis problemas de privacidade, dentre outros. Nossa proposta difere de outras por não utilizar cabeçalhos dos pacotes, ser baseada na coleta distribuída de dados (a partir de roteadores domésticos),

e ainda utilizar apenas uma pequena quantidade de informação coletada em roteadores domésticos. O modelo proposto pode ser implementado em cada roteador de maneira distribuída, com baixo custo computacional.

Propomos uma abordagem para a detecção de anomalias baseada em séries temporais de medições coletadas em roteadores domésticos de um ISP de porte médio. Nosso método utiliza a decomposição de tensores para detectar e diagnosticar eventos anômalos usando séries temporais multivariadas. A decomposição tensorial permite a extração do comportamento normal esperado para as métricas consideradas em diferentes intervalos de tempo, identificando a relação latente existente entre elas. Os resultados indicam que a abordagem é eficaz na detecção de eventos anômalos em dois cenários distintos que usamos como exemplo. Entretanto, enfatizamos que o método é geral e pode ser utilizado em outras aplicações.

O trabalho compartilha semelhanças com os anteriores de [Lakhina et al. 2004, Lakhina et al. 2005], onde um subespaço normal é definido aplicando o PCA e os residuais do modelo são usados para detectar anomalias em uma rede. Neste trabalho, a extração do subespaço normal é realizada com o modelo PARAFAC [Bro 1997], que naturalmente permite a decomposição de dados multidimensionais e preserva a relação entre as métricas em avaliação.

Como exemplo de utilização do método, inicialmente consideramos o problema de detecção de ataques DDoS. Mostramos que, a partir de um conjunto de medições simples e não intrusivas (sem inspeção de pacotes), utilizando apenas contadores de bytes e pacotes, o método proposto possui uma alta taxa de detecção. Portanto, uma aplicação do nosso trabalho é a sua utilização por ISPs para mitigar o impacto de ataques distribuídos. Uma segunda aplicação é a detecção de intervalos de degradação de desempenho a partir de medições realizadas a partir de roteadores domésticos. Neste caso, anomalias seriam tais intervalos. Este é um exemplo de aprendizado não supervisionado, a partir de medições de latência e perda de pacotes. O processo pode ser facilmente automatizado para identificar e localizar tais anomalias e analisar a qualidade da rede em partes distintas da topologia de um ISP.

Contribuições Nossas principais contribuições são resumidas no que se segue.

• *Decomposição de tensores para detecção de anomalias na rede.* Propomos um *framework* baseado na decomposição tensorial para detectar eventos anômalos em redes. Mostramos que o modelo PARAFAC fornece uma maneira interpretável e eficiente para extrair o comportamento normal esperado, considerando a correlação entre diferentes métricas.

• *Uso de dados reais coletados em roteadores domésticos.* Empregamos séries temporais obtidas a partir de dados reais de medições realizadas em roteadores domésticos. Um requisito essencial da proposta é a preservação da privacidade dos usuários. Dessa forma, mostramos que o nosso método é capaz de detectar diferentes tipos de anomalia a partir de métricas simples, sem realizar inspeção de pacotes.

• *Uso em aplicações distintas.* Mostramos como o mesmo método pode ser utilizado em diferentes aplicações, com bons resultados.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos

relacionados. Discutimos o método de decomposição tensorial na Seção 3. A Seção 4 mostra a abordagem proposta para detecção de anomalias. A aplicação do método para detectar ataques DDoS (Aplicação I) está na Seção 5. A Seção 6 descreve a detecção de eventos de degradação do desempenho da rede (Aplicação II) usando nossa abordagem. Nossas conclusões estão resumidas na Seção 7.

2. Trabalhos Relacionados

A grande maioria dos trabalhos da literatura utiliza inspeção de pacotes no núcleo da rede [Lakhina et al. 2004, Lakhina et al. 2005, Maruhashi et al. 2011, Xie et al. 2018, Silveira et al. 2011]. Um trabalho recente também emprega inspeção de pacotes, mas em roteadores residenciais [Doshi et al. 2018]. Nosso trabalho utiliza medições realizadas em roteadores domésticos, sem inspecionar pacotes, provendo uma estratégia simples, eficiente e que preserva a privacidade dos usuários.

Trabalhos recentes do nosso grupo também fizeram uso de medições em roteadores domésticos sem inspecionar pacotes [Mendonça et al. 2019, Santos et al. 2019, Streit et al. 2019]. O trabalho de [Mendonça et al. 2019] é focado em ataques DDoS que são detectados com alta precisão utilizando apenas estatísticas simples dos contadores de bytes e pacotes obtidas em uma janela de amostras. Por outro lado, o trabalho atual se utiliza de outro método (decomposição de tensores) e mostramos que ele pode ser empregado de forma abrangente para detectar diferentes tipos de anomalia. Além disso, mostramos que os resultados da nossa abordagem são interpretáveis, podendo-se inferir o comportamento normal diário de um usuário, um dos desafios para se detectar anomalia. O trabalho de [Santos et al. 2019] foca na detecção de pontos de mudança de padrão de medidas de QoS (*change point detection problem*). No entanto, o método requer a estimativa de três hiper-parâmetros, enquanto que o proposto apenas necessita da estimativa manual do número de *clusters*. Além disso, esse trabalho considera concomitantemente latência e perda de pacotes e, como consequência, é possível fazer uma análise mais refinada das anomalias em relação ao trabalho anterior.

Há trabalhos na literatura que empregam o método de extração do subespaço para a detecção de anomalias na rede. Com o método de extração do subespaço, eventos anômalos podem ser detectados (i) pelo modelo [Maruhashi et al. 2011, Koutra et al. 2012, Mao et al. 2014] ou (ii) pelos residuais [Lakhina et al. 2004, Lakhina et al. 2005, Sun et al. 2006, Callegari et al. 2011, Xie et al. 2018]. Na primeira categoria, o modelo é usado para identificar padrões latentes e recorrentes nos dados que possam indicar eventos de interesse. Na segunda abordagem, anomalias são detectadas a partir de residuais que se encontram fora do subespaço reconhecido como normal/recorrente. Abaixo detalhamos as diferenças em relação ao método proposto.

Detecção pelos padrões do modelo

Em [Koutra et al. 2012] os autores usam o método de PARAFAC para encontrar fatores latentes em seu modelo e identificar atividades de uso comum da rede e eventos que se assemelham a ataques a partir de um conjunto de dados com a estrutura (IP de origem $\times$ IP de destino $\times$ número da porta $\times$ *timestamp*). Diferentemente da nossa abordagem, eventos anômalos não são detectados automaticamente e requerem análise humana. Além disso, o dataset é obtido a partir da inspeção de pacotes. Da mesma forma, o trabalho [Maruhashi et al. 2011] sugere atividades suspeitas na rede (como var-

redura de portas e disseminação de worms) procurando subgráficos anormais de padrões identificados pela decomposição tensorial. O conjunto de dados tem o formato (IP de origem $\times$ IP de destino $\times$ *timestamp* ou número da porta). Nosso método não utiliza dados extraídos a partir de *headers* de pacotes, como IP e número de porta. Além disso, o método de [Maruhashi et al. 2011] também depende fortemente da escolha manual dos padrões considerados interessantes.

Detecção pelos residuais

Trabalhos anteriores consideram a utilização do PCA para a definição do subespaço normal [Lakhina et al. 2004, Lakhina et al. 2005]. O modelo obtido é utilizado para a extração de residuais a partir dos quais anomalias podem ser identificadas. Porém, o algoritmo do PCA é muito sensível à dimensionalidade do subespaço normal [Ringberg et al. 2007] . A seleção do número apropriado de dimensões torna-se surpreendentemente difícil, o que pode ter uma influência significativa na taxa de falsos positivos. Neste trabalho consideramos a utilização do PARAFAC para determinar o subespaço normal a partir dos quais os residuais são extraídos. O modelo PARAFAC possui a propriedade de solução única e permite identificar fatores comuns e intrínsecos nos dados sem a necessidade de qualquer outro método externo de rotação (por exemplo, Varimax) [Harshman and Lundy 1984, Kruskal 1983]. Isso é especialmente vantajoso, pois (no PCA) diferentes técnicas de rotação podem dar origem a diferentes conclusões sobre a estrutura do subespaço normal. Como resultado, no PARAFAC torna-se mais fácil definir o número apropriado de fatores necessários para delimitar o subespaço normal.

O trabalho de [Xie et al. 2018] propõe um método de detecção de anomalias utilizando um modelo PARAFAC modificado que considera características não lineares dos dados. O algoritmo proposto considera a similaridade entre fatias de cada modo do tensor durante o processo de treinamento. Além disso, as anomalias são associadas a um segundo tensor esparso que não é utilizado durante o processo de otimização. Uma desvantagem deste trabalho é a avaliação do método apenas utilizando dados de ataques gerados artificialmente, a partir de uma distribuição de probabilidade arbitrária. Nosso trabalho considera um conjunto de dados real obtidos a partir de medições realizadas em roteadores residenciais. Além disso, os autores não aplicam técnicas de validação para determinar o número apropriado de fatores, nem consideram a interpretabilidade do modelo, uma das principais vantagens do método PARAFAC. Nosso trabalho utiliza a técnica *Split-Half Validation* para a escolha do número de fatores e analisa o comportamento normal obtido a partir da decomposição fatorial.

3. Decomposição de Tensores

Nesta seção, fornecemos uma base teórica sobre decomposição de tensores e descrevemos nossa notação. Um tensor é uma matriz multidimensional, denotada por $\mathcal{X}$. Geralmente nos referimos às dimensões de $\mathcal{X}$ como *modos*. Consideramos um tensor de terceira ordem $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ dado pela soma dos produtos externos de três vetores, ou seja,

$$\begin{aligned} \mathcal{X} &= \mathcal{M} + \mathcal{E} \\ \mathcal{M} &= \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad \mathbf{a}_r \in \mathbb{R}^I, \mathbf{b}_r \in \mathbb{R}^J, \mathbf{c}_r \in \mathbb{R}^K \end{aligned} \tag{1}$$

Os residuais são indicados por $\mathcal{E}$, enquanto o número de fatores é definido por R. As *matrizes fatoriais* (ou *loadings*) definem o modelo $\mathcal{M}$:

$$A = [\mathbf{a}_1\mathbf{a}_2 \dots \mathbf{a}_R] \in \mathbb{R}^{I\times R},\ B = [\mathbf{b}_1\mathbf{b}_2 \dots \mathbf{b}_R] \in \mathbb{R}^{J\times R},\ C = [\mathbf{c}_1\mathbf{c}_2 \dots \mathbf{c}_R] \in \mathbb{R}^{K\times R}.$$

Para resolver o PARAFAC, minimizamos a soma dos quadrados dos resíduos, ou seja, a diferença entre $\mathcal{X}$ e $\mathcal{M}$. Essa é uma função não-convexa; no entanto, se fixarmos duas das matrizes fatoriais, o problema é reduzido a uma regressão linear de mínimos quadrados para a terceira matriz. Esse é o procedimento de mínimos quadrados alternados (*Alternating Least Squares* - ALS) [Bro 1997]. No algoritmo ALS as matrizes fatoriais são estimadas uma de cada vez, mantendo as outras fixas. O processo é repetido até que um critério de convergência seja satisfeito ou até que não ocorra mudança nas estimativas.

Consideramos neste trabalho o uso do método *Split-Half Validation* (SV) [Harshman 1984] em combinação com o *Tucker Congruence Coefficient* (TCC) [Lorenzo-Seva and Ten Berge 2006] para determinar R e julgar se a solução é única e generalizável para outro conjunto de dados semelhante.

4. Framework

Nesta seção discutimos o *framework* proposto neste trabalho. A metodologia proposta é composta pelas seguintes etapas:

1. **Pré-processamento:** Nesta etapa são feitas as trasformações nos dados necessárias à decomposição tensorial, tais como mudanças na escala dos dados e filtragem.
2. **Decomposição tensorial:** Nesta etapa, aplicamos a decomposição no tensor para extração do subespaço normal. Utilizamos o método PARAFAC devido à garantia de solução única e a sua capacidade de lidar com dados multivariados.
3. **Extração residual:** Nesta etapa, empregamos o modelo obtido pela decomposição tensorial para extrair os residuais e realizar a detecção de anomalias. A idéia é explorar o fato de que as anomalias não são bem modeladas pelo subespaço normal, permitindo a separação entre comportamento normal e anômalo através da análise do residual.
4. **Classificação/clusterização de anomalias:** O estágio final varia de acordo com a aplicação considerada. Quando todas as anomalias no conjunto de dados são conhecidas, realizamos uma classificação supervisionada. Por outro lado, existem aplicações em que os rótulos para as anomalias são desconhecidos ou difíceis de se obter. Para esses casos, consideramos uma abordagem não supervisionada baseada em clusterização.

Os métodos de decomposição tensorial são descritos na Seção 3. Detalhes sobre a extração residual são dados a seguir. A Seção 5 apresenta um exemplo de classificação de anomalias, enquanto a Seção 6 descreve uma aplicação do método de clusterização.

Extração residual [Bro 1997]

Nossa técnica de detecção de anomalias é baseada na análise de residuais obtidos a partir do modelo de decomposição tensorial. Extrai-se o comportamento normal das medições usando a decomposição dos fatores e detecta-se as anomalias através da análise de desvios dos padrões modelados.

Dividimos os dados dos usuários em séries temporais diárias com granularidade de um minuto. São consideradas diferentes métricas de interesse obtidas durante o monitoramento da rede. Denotamos cada série como um par usuário-dia, também chamado

de par UD. Expressamos I como o número de pares UD em nosso conjunto de dados. Neste artigo, trabalhamos com tensores de entrada que compreendem três modos, a saber, pares usuário-dia (modo A), medidas de interesse (modo B) e minutos (modo C), indicados pelos índices i, j e k, respectivamente. Formalmente, para cada UD com medidas $\mathcal{X}_i \in \mathbb{R}^{1\times J\times K}$, obtemos um modelo $\mathcal{M}_i \in \mathbb{R}^{1\times J\times K}$ usando o PARAFAC, onde $\mathcal{X}_i$ representa a i-ésima fatia horizontal do tensor $\mathcal{X}$. Como um UD se refere a um dia, $K = 1440$.

Na análise não supervisionada consideramos os residuais obtidos durante o processo de treinamento do modelo PARAFAC. Os residuais são medidos a partir da diferença entre o modelo e o dataset de entrada $\mathcal{E}_i = \mathcal{X}_i - \mathcal{M}_i$, onde $\mathcal{E}_i \in \mathbb{R}^{1\times J\times K}$. No ambiente supervisionado, o modelo PARAFAC é obtido a partir de um conjunto de treinamento $\mathcal{X}$ contendo séries temporais com comportamento normal. Desejamos detectar anomalias em um novo conjunto de UDs que não é utilizado no treinamento. Seja $\tilde{\mathcal{X}}_\kappa$ as medidas de um novo UD$_\kappa$. Utilizamos o modelo previamente treinado $\mathcal{M}$ (Equação 1) para obter o vetor $\tilde{\mathbf{a}}_\kappa$ do modo A. Calcula-se $\tilde{\mathbf{a}}_\kappa$ de forma a minimizar o erro quadrático. Para cada candidato à solução $\hat{\mathbf{a}}_\kappa$ o erro correspondente é dado por $\hat{E}_{\kappa(1)}$. Então,

$$\begin{aligned}\tilde{\mathcal{M}}_k = \tilde{\mathcal{X}}_k - \tilde{\mathcal{E}}_k &\Rightarrow \hat{\mathbf{a}}_\kappa (C \odot B)^T = \tilde{X}_{\kappa(1)} - \hat{E}_{\kappa(1)} \\ &\Rightarrow \tilde{\mathbf{a}}_\kappa (C \odot B)^T ((C \odot B)^T)^\dagger = \tilde{X}_{\kappa(1)} ((C \odot B)^T)^\dagger \\ &\Rightarrow \tilde{\mathbf{a}}_\kappa = \tilde{X}_{\kappa(1)} ((C \odot B)^T)^\dagger,\end{aligned}$$

onde a operação $\odot$ refere-se ao produto de Khatri-Rao das matrizes fatoriais B e C do modelo $\mathcal{M}$. A matriz $\tilde{X}_{\kappa(1)} \in \mathbb{R}^{1\times JK}$ é resultado da *matricização* no primeiro modo (A) de $\tilde{\mathcal{X}}_\kappa \in \mathbb{R}^{1\times J\times K}$. O símbolo sobrescrito † indica a Moore-Penrose pseudo-inversa aplicada a $(C \odot B)^T$. Dessa forma, os residuais de UD$_\kappa$ são obtidos por $\tilde{\mathcal{E}}_\kappa = \tilde{\mathcal{X}}_\kappa - \tilde{\mathcal{M}}_\kappa$, onde o modelo $\tilde{\mathcal{M}}_\kappa \in \mathbb{R}^{1\times J\times K}$ contém o novo vetor de fatores $\tilde{\mathbf{a}}_\kappa$.

5. Aplicação I: Detecção de ataques DDoS

Esta seção apresenta a aplicação de nossa metodologia para detectar ataques DDoS a partir de dispositivos domésticos. Diferentes tipos de vetores de ataques (ver Tabela 1) são gerados no laboratório (usando o código de *malwares* reais) e adicionados aos dados reais dos usuários em intervalos de tempo escolhidos aleatoriamente, empregando a metodologia de [Mendonça et al. 2019]. Portanto, contamos com uma abordagem supervisionada, pois sabemos exatamente quando os ataques são acionados.

Tabela 1. Tipos de ataques DDoS avaliados neste trabalho

Malware	Tipo de ataque (tamanho do payload)
Mirai	UDP flood (1400B)
Mirai	TCP SYN flood (0B)
Mirai	TCP ACK flood (0B)
Mirai	UDP PLAIN flood (1400B)
BASHLITE	UDP flood (1400B)
BASHLITE	TCP SYN flood (0B)
BASHLITE	TCP ACK flood (0B)

Tabela 2. Quantidade de pares UD em conjuntos diferentes

	Treinamento	Teste
D_1 e D_2: UDs sem ataques	$\|Tr_1\| = 11971$	$\|Te_1\| = 3331$
D_1: UDs com muitos ataques	$\|Tr_2\| = 1131$	$\|Te_2\| = 172$
D_2: UDs com um ataque	$\|Tr_2\| = 83407$	$\|Te_2\| = 27576$

5.1. Pré-processamento

Coletamos a cada minuto as taxas de bytes e de pacotes baixados (download) e enviados (upload) por um determinado usuário em um determinado dia (ou seja, par UD). A partir desses dados obtemos as séries temporais utilizadas como entrada para o método de decomposição tensorial.

Avaliamos dois datasets diferentes: o Dataset 1 (D_1) e o Dataset 2 (D_2). D_1 contem UDs que possuem em média 320 ataques por dia (a média considera todas as séries de 24 horas de todos os usuários). D_2 possui UDs com apenas um único ataque por dia. Para cada um desses datasets definimos dois conjuntos de treinamento: Tr_1 contem apenas séries sem nenhum ataque; Tr_2 possui UDs com ataques distribuídos de acordo com cada um dos datasets. Aplicamos o método PARAFAC para treinamento usando Tr_1 e obtemos o modelo $\mathcal{M}$. A partir de $\mathcal{M}$ calculamos $\tilde{\mathcal{M}}$ com o conjunto de UDs de Tr_2. Os classificadores são treinados com os residuais de Tr_2. Também dividimos os datasets em dois conjuntos de testes: Te_1 possui UDs sem ataques e Te_2 contem UDs com ataques combinados. Ambos são usados para avaliar a classificação de residuais. Os conjuntos de treinamento foram coletados entre 23 e 31 de março de 2019, enquanto os conjuntos de testes foram obtidos entre 2 e 4 de abril de 2019. A Tabela 2 detalha a quantidade de UDs em cada um desses conjuntos. Antes da aplicação da decomposição tensorial convertemos as medidas para escala logarítmica. Em seguida, aplicamos a normalização Min-Max em cada métrica de tráfego. Ao manter as métricas de tráfego em uma escala semelhante, capturamos as correlações entre elas e garantimos que tenham o mesmo impacto no processo de otimização.

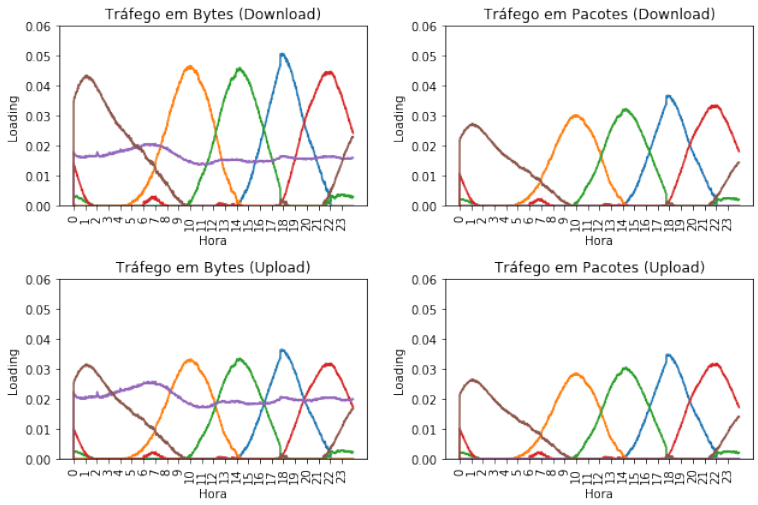
5.2. Decomposição tensorial

Nosso tensor é composto de três modos: (UD $\times$ métrica de tráfego $\times$ minuto). As métricas de tráfego correspondem às taxas de bytes/pacotes baixados/enviados a cada minuto por usuários residenciais. Construímos o tensor usando o conjunto de treinamento Tr_1, onde $|Tr_1| = 11971$. Portanto, obtemos um tensor de terceira ordem $\mathcal{X} \in \mathbb{R}^{11971 \times 4 \times 1440}$. Aplicamos o método *Split-Half Validation* e definimos o número de fatores $R = 6$. Essa é a quantidade máxima de fatores validada pelo *Split-Half Validation*.

A Figura 1(a) apresenta os fatores do modo minuto ponderados pelos *loadings* associados às medições das taxas de bytes, enquanto a Figura 1(b) mostra os fatores ponderados pelos loadings das taxas de pacotes. Nas duas figuras, plotamos gráficos separados para upload e download. Um dos fatores (representado em roxo) não está fortemente associado aos tempos de uso da rede - geralmente é constante ao longo do dia. Os fatores restantes identificam maior uso da rede em diferentes períodos do dia. Além disso, a diferença de escala entre o número de bytes baixados e enviados é maior que a diferença para o número de pacotes baixados e enviados. Isso indica que as conexões trocam um número semelhante de pacotes de download e upload, mas os pacotes de upload geralmente carregam menos dados.

5.3. Extração residual

Extraímos residuais para os conjuntos Tr_2, Te_1 e Te_2 de cada dataset (D_1 e D_2) usando a técnica de extração residual descrita na Seção 4. Treinamos diferentes classificadores



(a) Fatores associados às taxas de bytes

(b) Fatores associados às taxas de pacotes

Figura 1. Aplicação I: Fatores obtidos pelo PARAFAC

com os residuais obtidos pelo conjunto Tr_2 e os testamos com os residuais dos conjuntos Te_1 e Te_2. Consideramos o problema de detectar ataques a cada minuto para cada usuário. As entradas fornecidas aos classificadores são os residuais de todas as métricas de tráfego para cada minuto e para cada UD e, portanto, as amostras de entrada têm quatro medidas.

5.3.1. Resultados da classificação

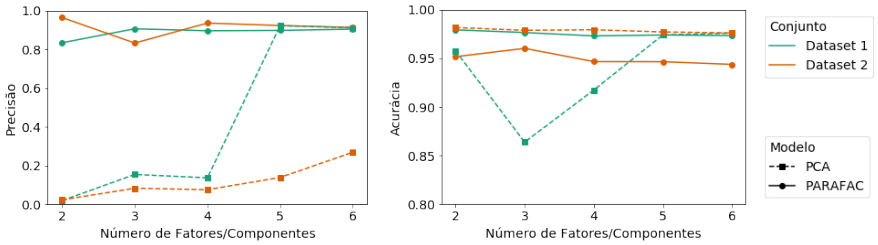
Para estimar a capacidade de detecção de ataques consideramos os seguintes classificadores: *Regressão Linear*, *Regressão Logística*, *Árvore de Decisão*, *Random Forest* e *Gaussian Naive Bayes*. Consideramos duas métricas para avaliação: Acurácia e Precisão. A Acurácia mede a porcentagem de anomalias detectadas. O valor desta métrica é dado por $\frac{n_d}{n}$, onde n_d é o número de ataques detectados, e n é o número total de ataques no conjunto Te_2. Consideramos que um ataque é detectado se for identificada uma anomalia em pelo menos um dos *slots* de tempo no intervalo de duração do ataque. A Precisão é calculada da seguinte forma: $\frac{\text{VP}}{\text{VP+FP}}$, onde VP (Verdadeiros Positivos) é o número de minutos onde um ataque é detectado e houve um ataque, e FP (Falsos Positivos) é o número de minutos onde um ataque é detectado e não houve um ataque. A Precisão tem valores mais baixos quando o valor de FP aumenta. A Tabela 3 apresenta os resultados obtidos usando o Dataset 1.

Tabela 3. Resultados dos classificadores para o Dataset 1

Classificadores	Precisão	Acurácia
Regressão Linear	$0,9460$	$0,9324$
Regressão Logística	$0,8143$	$0,9789$
Árvore de Decisão	$0,9037$	$0,9732$
Random Forest	$0,7585$	$0,9840$
Gaussian Naive Bayes	$0,3066$	$0,9721$

A Tabela 3 indica que os classificadores mais promissores são *Regressão Linear* e *Árvore de Decisão*. Em certos cenários a detecção incorreta de um ataque pode ser extremamente prejudicial. Por exemplo, um cliente legítimo pode ter sua conexão bloqueada se o classificador falsamente declarar um ataque. Nesse caso, o classificador mais adequado seria *Regressão Linear*. Por outro lado, em cenários onde o impacto do ataque é muito grande, a utilização de um classificador de maior acurácia, como a *Árvore de Decisão*, é recomendada.

Com o objetivo de comparar o desempenho do nosso método com outros da literatura baseados no PCA, usamos como entrada para o classificador *Árvore de Decisão* os residuais obtidos pelos modelos PARAFAC e PCA de ambos os datasets D_1 e D_2 e variamos a quantidade de fatores (ou componentes, no caso do PCA) no intervalo $[2, 6]$. A Figura 2 indica a robustez e melhor desempenho do PARAFAC em comparação ao PCA. As linhas pontilhadas (contínuas) apresentam os valores obtidos para a Precisão e Acurácia do PCA (PARAFAC) para ambos os datasets quando variamos o número de componentes (fatores). Nota-se que para o dataset D_2 (um ataque por UD), a Precisão do PCA é inferior a $0,26$ enquanto que para o dataset D_1 (vários ataques por UD) a Precisão depende fortemente do número de componentes. Já o método PARAFAC apresenta uma Precisão contida no intervalo $(0,83; 0,96)$ para ambos os datasets, ou seja, apresenta pouca variação com o número de fatores. Para a métrica Acurácia, os resultados de ambos os métodos são similares para o dataset D_2. Para o dataset D_1, o PCA apresenta valores que variam com o número de componentes. Os resultados apresentados demonstram que o método PARAFAC possui maior robustez e melhor desempenho que o PCA e sugerem o potencial da abordagem de decomposição tensorial na detecção de ataques originados em redes domésticas.



(a) Precisão (b) Acurácia

Figura 2. *Árvore de Decisão* aplicada nos residuais do PARAFAC e do PCA

6. Cenário II: Detecção de períodos de degradação do desempenho da rede

Nesta seção discutimos a aplicação da nossa metodologia para detectar eventos de degradação de desempenho na rede de um ISP. É difícil definir um conjunto de anomalias de rede [Lakhina et al. 2005]. Na ausência de rótulos confiáveis para identificá-las, utilizamos a clusterização dos residuais do modelo de decomposição tensorial para agrupar eventos com comportamento similar.

6.1. Pré-processamento

Neste cenário aplicamos o *framework* proposto utilizando séries temporais de latência e perda com granularidade de um minuto em 2.562 roteadores domésticos durante um mês

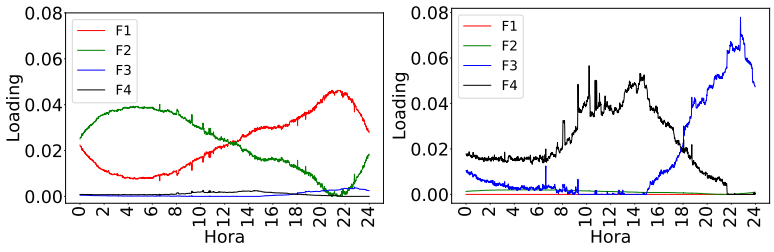
(setembro de 2018). As séries são obtidas a partir do envio de um trem de 100 pacotes ICMP com intervalos de 10 milissegundos destinado a um servidor localizado na rede do ISP. O resultado dessas medições pode ser afetado pelo tráfego concorrente gerado pelo usuário [Sundaresan et al. 2011]. Portanto, filtramos amostras cujo *cross-traffic* é maior que um limite $\theta = 2,5$ Mbps. Após essa filtragem, consideramos em nossa análise apenas séries temporais com pelo menos $\eta = 1000$ amostras de medição.

Períodos de indisponibilidade podem ser detectados a partir da falta de amostras durante vários *slots* de tempo. Portanto, codificamos cada minuto sem resultados de medição de perda como uma amostra de indisponibilidade, representada por uma fração de perda igual a 1. Como o método PARAFAC é capaz de lidar com *missing data* durante o processo de otimização, representamos os intervalos de tempo com indisponibilidade no servidor e *cross-traffic* como dados faltantes. Por fim, consideramos a escala logarítmica para a aplicação do método de decomposição tensorial. Foram utilizadas 57.955 séries temporais multivariadas com valores de latência e perda a cada minuto.

6.2. Decomposição tensorial

Modelamos os dados de medição usando um tensor com três modos (UDs $\times$ métrica de desempenho $\times$ minuto). Como usamos séries temporais diárias de medições de latência/perda como entradas, obtemos um tensor de terceira ordem $\mathcal{X} \in \mathbb{R}^{57955 \times 2 \times 1440}$. Usamos o método *Split-Half Validation* para definir o número de fatores $R = 4$.

A Figura 3(a) apresenta os fatores do modo minuto ponderados pelos *loadings* associados às medições de latência. É possível observar que o comportamento diário das medidas de latência possui dois fatores dominantes: verde e vermelho. O primeiro (F2) representa picos observados pela manhã, enquanto o segundo (F1) representa latências mais altas à noite, especialmente durante o período com maior quantidade de tráfego (por volta das 22h). Quando consideramos os loadings associados à perda, representados na Figura 3(b), observamos que os dois fatores menos usados para representar a latência (F3 e F4) são os mais utilizados para descrever os padrões diários de perda. O fator F4 explica períodos de maiores perdas durante a manhã e a tarde, enquanto o fator F3 representa congestionamentos observados à noite.



(a) Fatores associados à latência (b) Fatores associados à perda

Figura 3. Cenário II: Fatores obtidos pelo PARAFAC

6.3. Extração residual

Para a detecção de períodos de degradação de desempenho da rede, consideramos métricas obtidas a partir dos residuais de latência, dos residuais de perda e dos residuais de perda desconsiderando amostras com perda igual a 100%. Cada métrica pode ser usada

para detectar um tipo diferente de anomalia. Por exemplo, latência e perdas menores que 100% podem detectar períodos com alto congestionamento, enquanto amostras de perda iguais a 100% podem detectar falhas nos links da rede. Para facilitar a interpretação, extraímos três estatísticas simples para cada métrica: média, desvio padrão e 95^o percentil, totalizando 9 atributos.

6.4. Clusterização de anomalias

Resultados da clusterização

Consideramos o algoritmo K-Means para a clusterização dos residuais devido à sua simplicidade e interpretabilidade. Para a escolha do número de clusters, utilizamos o método do cotovelo (*Elbow Method*) comumente usado na literatura [Lakhina et al. 2005]. Utilizamos cinco clusters, uma vez que um número maior de clusters resulta em grupos com comportamento similar. Anteriormente à fase de agrupamento, aplicamos a normalização z-score para evitar que diferenças na escala dos atributos afetem os resultados.

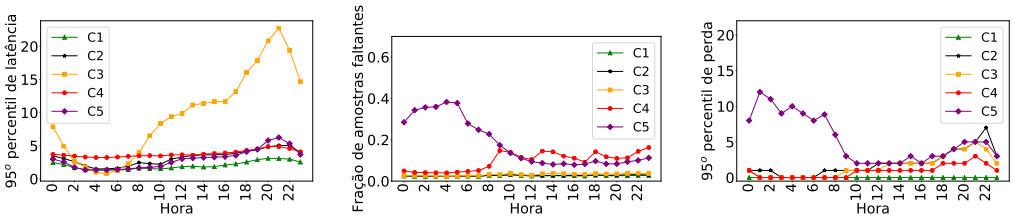
Investigamos o significado do agrupamento obtido sumarizando todas as séries temporais atribuídas a cada cluster para cada métrica (latência, perda e indisponibilidade). Para avaliar a latência por cluster, subtraimos cada amostra da série temporal diária de latência pelo seu valor mais baixo. Assim inferimos o tempo de enfileiramento dos pacotes durante períodos de congestionamento.

A Figura 4(a) mostra o 95^o percentil das amostras de latência normalizadas por hora para cada cluster. O cluster C3 agrupa séries temporais de usuários com os atrasos elevados especialmente após as 18h. A Figura 4(b) exibe a fração de amostras de indisponibilidade por hora para cada cluster. O cluster C5 contém séries temporais de usuários em um dia (UD) onde há períodos de alta indisponibilidade, especialmente até às 8h. O cluster C4 inclui usuários-dia (UD) com significativas amostras de indisponibilidade após 9h da manhã. A Figura 4(c) apresenta o 95^o percentil das amostras de perda por hora para cada cluster. Neste caso, o cluster C5 indica que há usuários-dia com perdas significativas até 8h da manhã. O cluster C2 inclui séries com altas perdas à noite, especialmente às 22h. Por fim, observa-se que o cluster C1 apresenta séries com valores baixos para todas as métricas consideradas.

Com base na sumarização das séries temporais, definimos uma interpretação para cada cluster. O cluster C1 representa séries temporais (usuários-dia) com boa qualidade de rede (baixa latência, baixa perda, baixa indisponibilidade). O cluster C2 agrupa séries temporais com altas perdas e baixa indisponibilidade. O cluster C3 agrupa séries temporais com alta latência. O cluster C4 agrupa séries com alta indisponibilidade e perdas moderadas. Finalmente, o cluster C5 contém séries temporais com alta indisponibilidade e alta taxa de perdas, especialmente até às 8h.

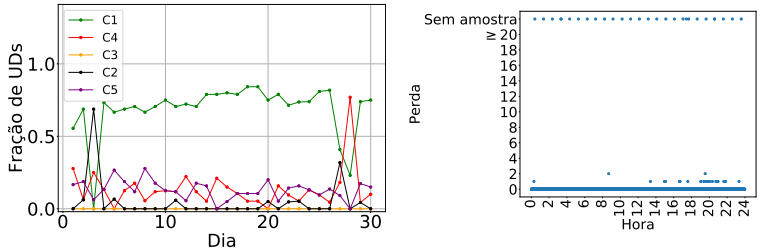
Correlação espacial

Para identificar períodos de degradação de desempenho que afetam múltiplos usuários próximos geograficamente, correlacionamos espacialmente os resultados da clusterização dos residuais utilizando informação sobre a topologia da rede. Espera-se que clientes com localização próxima apresentem desempenho parecido, uma vez que rotas similares são utilizadas. Primeiramente, agrupamos manualmente clientes que se encontram em uma mesma região da topologia do ISP. Em seguida, analisamos para cada grupo a fração de usuários atribuídos a cada cluster em cada dia do dataset.

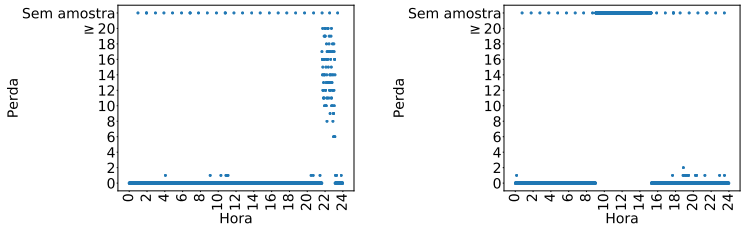


(a) Latência (b) Indisponibilidade (c) Perda

Figura 4. Sumarização das séries temporais pertencentes a cada cluster



(a) Fração de usuários por cluster por dia (b) Usuário no cluster C1 (04/09)

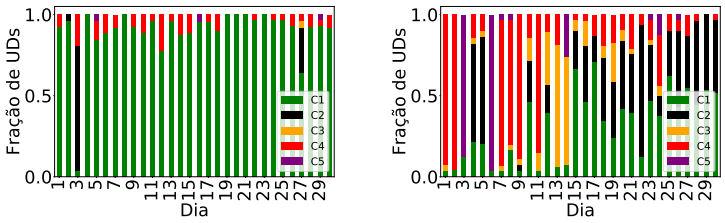


(c) Usuário no cluster C2 (03/09) (d) Usuário no cluster C4 (28/09)

Figura 5. Exemplo de correlação espacial

Exemplificamos os resultados da correlação espacial utilizando uma região específica da rede. Resultados similares são encontrados em diferentes partes da topologia. A Figura 5(a) mostra a fração diária de UDs por cluster. Um dia de comportamento comum da rede é exemplificado na Figura 5(b): a maioria dos usuários é associada ao cluster C1 e poucas perdas são observadas. Por outro lado, a Figura 5(c) apresenta um dia em que um período de degradação de desempenho ocorre: um grande número de usuários é associado ao cluster C2, em que as séries temporais possuem altos valores de perda entre 22 e 23 horas. Outro tipo de evento detectado pela correlação espacial é apresentado na Figura 5(d). Neste dia, um período de indisponibilidade entre 9 e 15 horas é observado em múltiplos usuários.

Os resultados da aplicação do *framework* podem resumir a qualidade de cada região da rede com base na fração de séries temporais atribuídas ao cluster de bom desempenho (cluster C1). A Figura 6 apresenta os resultados da clusterização em duas partes diferentes da rede. Observa-se que uma região frequentemente apresenta uma alta fração de usuários com bom desempenho (Figura 6(a)), embora eventos de degradação sejam identificados nos dias 03/09 e 22/09. Ao mesmo tempo, a Figura 6(b) mostra uma região



(a) Região A (b) Região B

Figura 6. Sumário do desempenho da rede obtido pela clusterização

em que uma fração menor de usuários é atribuída ao cluster C1.

7. Conclusão

Neste trabalho, propomos um *framework* baseado em decomposição tensorial para detectar anomalias na rede. Aplicamos o método PARAFAC e extraímos os residuais obtidos pelo modelo com o objetivo de identificar comportamento anormal. Mostramos a flexibilidade do nosso método, usando duas aplicações como exemplo. Primeiro, consideramos a detecção de ataques DDoS usando técnicas supervisionadas. Os resultados mostram que podemos obter valores altos para acurácia e precisão usando diferentes classificadores. Além disso, o nosso método apresenta melhor desempenho e robustez quando comparado com o PCA. Em seguida, usamos a metodologia proposta para identificar períodos de degradação do desempenho da rede através de uma abordagem não supervisionada. O método mostra-se capaz de identificar períodos de degradação que afetam vários clientes e analisar o desempenho de diferentes partes da topologia de um ISP.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Este trabalho é parcialmente suportado por projeto de cooperação MCTIC-RNP/NSF, MCTIC/FAPESP, e ainda por projetos do CNPq e FAPERJ.

Referências

Bro, R. (1997). Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171.

Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., and Pepe, T. (2011). A novel pca-based network anomaly detection. In *IEEE ICC 2011*, pages 1–5.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Doshi, R., Apthorpe, N., and Feamster, N. (2018). Machine learning DDoS detection for consumer IoT devices. *IEEE Security and Privacy Workshops*, pages 29–35.

Fadilpašić, S. (2019). Aws hit by DDoS attack. Acessado em 19/12/2019.

Harshman, R. A. (1984). "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*, pages 566–591.

Harshman, R. A. and Lundy, M. E. (1984). The parafac model for three-way factor analysis and multidimensional scaling. *Research methods for multimode data analysis*, 46:122–215.

Koutra, D., Papalexakis, E. E., and Faloutsos, C. (2012). Tensorsplat: Spotting latent anomalies in time. In *16th Panhellenic Conference on Informatics*, pages 144–149.

Kruskal, J. (1983). Multilinear methods. In *Proc. Symp. Appl. Math*, volume 28, page 75.

Lakhina, A., Crovella, M., and Diot, C. (2004). Diagnosing network-wide traffic anomalies. In *ACM computer communication review*, volume 34, pages 219–230.

Lakhina, A., Crovella, M., and Diot, C. (2005). Mining anomalies using traffic feature distributions. In *ACM computer communication review*, volume 35, pages 217–228.

Lorenzo-Seva, U. and Ten Berge, J. M. (2006). Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology*, 2(2):57–64.

Mao, H.-H., Wu, C.-J., Papalexakis, E. E., Faloutsos, C., Lee, K.-C., and Kao, T.-C. (2014). Malspot: Multi 2 malicious network behavior patterns analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 1–14.

Maruhashi, K., Guo, F., and Faloutsos, C. (2011). Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 203–210.

Mendonça, G., Santos, G., de Souza e Silva, E., Leão, R., Menasché, D., and Towsley, D. (2019). An extremely lightweight approach for ddos detection at home gateways. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5012–5021.

Ringberg, H., Soule, A., Rexford, J., and Diot, C. (2007). Sensitivity of pca for traffic anomaly detection. In *ACM SIGMETRICS Performance Evaluation Review*, volume 35, pages 109–120.

Santos, G. H., Mendonça, G., de Souza, E., Leão, R. M. M., Menasche, D. S., et al. (2019). Análise não supervisionada para inferência de qualidade de experiência de usuários residenciais. In *SBRC 2019*, pages 958–971.

Silveira, F., Diot, C., Taft, N., and Govindan, R. (2011). Astute: Detecting a different class of traffic anomalies. *ACM SIGCOMM CCR*, 41(4):267–278.

Streit, A. G., Leão, R. M. M., de Souza, E., Menasche, D., et al. (2019). Descobrindo perfis de tráfego de usuários: uma abordagem não supervisionada. In *SBRC 2019*, pages 169–182.

Sun, J., Tao, D., and Faloutsos, C. (2006). Beyond streams and graphs: dynamic tensor analysis. In *12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383.

Sundaresan, S., de Donato, W., N.Feamster, Teixeira, R., Crawford, S., and Pescapè, A. (2011). Broadband internet performance: A view from the gateway. In *ACM SIGCOMM 2011*.

Xie, K., Li, X., Wang, X., Xie, G., Wen, J., and Zhang, D. (2018). Graph based tensor recovery for accurate internet anomaly detection. In *IEEE INFOCOM 2018*, pages 1502–1510.