

# Análise de dados de log de tráfego de rede: agrupamentos para caracterizar Slices - Log2SLA

1<sup>st</sup> Marco Antonio Firmino de Sousa

*Redes e Comunicação de Dados*

*Universidade de São Paulo*

São Paulo, Brasil

marco.volare@gmail.com

**Resumo**—A alocação de recursos da rede de dados de forma automatizada e com garantias de qualidade de serviços e experiência, mesmo em um cenário de demandas contraditórias entre os usuários, possui solução técnica com a adoção do *Network Slice* por meio de Inteligência Artificial. O presente trabalho, abordou um processo completo de análise de dados para validar o cenário propício para a produção de modelos de aprendizado de máquina.

**Index Terms**—5G, Slice Network, Inteligência Artificial

## I. INTRODUÇÃO

A rede 5G possui características disruptivas desde sua concepção [1], [2]. Planejada para atender simultaneamente usos contraditórios e competitivos entre si, implementa um recurso técnico que apoia a rede a atender satisfatoriamente a demanda crescente. A capacidade de fatiar a rede garantindo isolamento e confiabilidade é suportada pelo dinamismo de uma estrutura primordialmente baseada em software. O *Network Slicing* é uma técnica que permite a multialocação de recursos para instanciar uma rede virtual fim-a-fim garantindo o isolamento, separação lógica e física de recursos.

A alocação de recursos para a composição de um *slice* deve suportar à função de gerenciamento do ciclo de vida do *slice* [3], que por sua vez, deve atender aos requisitos especificados no SLA e regras de QoS e QoE em atendimento à métricas de latência, taxa de transferência, capacidade dos recursos disponíveis [4]. Considerando a característica de computação distribuída, sistemas e funções virtuais, a dinâmica variável da rede, bem como a dinamicidade e variância dos parâmetros ao longo do tempo tornam a orquestração de recursos para a composição de *slice* uma tarefa complexa e apropriada para a aplicação de solução baseada em Inteligência Artificial (IA). Todavia, uma premissa para a aplicação de técnicas de IA é a existência de um modelo matemático, o conhecimento de um especialista ou um conjunto de dados. No contexto deste trabalho, dada a inexistência das duas primeiras opções, torna-se necessário construir ou buscar por *dataset* que retratem as métricas de eficiência de *slices*, ou ainda utilizar streaming de dados para a aplicação em modelos de IA.

Desta forma, o monitoramento de dados do tráfego de entrada da rede ou histórico de comportamento dos *slices* em forma de dados brutos por *streaming* permitem representar o comportamento real da rede em termos de QoS e servir de base para a obtenção de modelos ótimos. Em geral, o interesse de

pesquisa em alocação de recursos concentra-se principalmente em como implementar instâncias de *slice* de acordo com a descrição dos requisitos de recursos [5] e atendimento a acordo de serviços.

O propósito deste trabalho é validar uma metodologia para coleta e análise de dados de rede flexível suficiente para comportar o modelo iterativo e iterativo para a produção de Modelos de Machine Learning [6]. Como resultado inicial, espera-se estabelecer um formato de dado para permitir o treinamento de diversos modelos de IA e realizar o reconhecimento e recomendação de *slice* dado o fluxo gerado pelo cliente.

## II. TRABALHOS RELACIONADOS

Em geral, o interesse de pesquisa em alocação de recursos concentra-se principalmente em como implementar instâncias virtuais de rede sob recursos físicos compartilhados. O fatiamento de rede fim-a-fim para rede móvel 5G com utilização de ML tem sido intensamente estudado [3], [4], [7]–[11]. Uma das questões mais importantes ao relacionar a utilização de *slice* por atribuição realizada por aprendizado de máquina é garantir geração de modelos ótimos, para que desta forma evite-se, simultaneamente, a subutilização de recursos, como também a quebra de SLA.

Estabelecer métricas e um mecanismo de recuperação de fatia ideal para demandas determinísticas é o ponto de partida implementada por Wen [12], onde buscou-se construir a solução ideal para utilizar como referência em outros algoritmos. Em demandas estocásticas a opção de mecanismos de otimização que possui, geralmente, convergência lenta é adotada inicialmente, optando pela utilização de machine learning como solução futura. Estabelecer um conjunto de dados que refletem uma solução ótima para balizar o aprendizado de máquina é um cenário apropriado para a construção de modelos de aprendizado supervisionado. Todavia, a diversidade de cenários e clientes demandantes das redes tornam a definição padronizada uma tarefa complexa.

Desta forma, explorar tipos de aprendizado não dependentes de *dataset* contendo o rótulo classificador de modelo ótimo, apresenta-se como alternativa plausível e viável. Assim, por exemplo, a utilização de aprendizado não-supervisionado, modelos de redes neurais recorrentes e convolucionais são adotadas na construção de modelos para soluções relacionadas

ao fatiamento de rede. No trabalho proposto por Toscano [13], as redes neurais do tipo *Long-Short Term Memory* (LSTM) são utilizadas para o provisionamento de *slice* em um mecanismo simples dotado de apenas quatro parâmetros para a compor o SLA, sendo ( $R_s$ ,  $K_s$ ,  $W_s$  e  $D_s$ ), onde  $R_s$  é a proporção média de utilização de recursos, sendo que  $R_s \in (0, 1]$ . Para garantir uma tolerância ao operador da rede,  $K_s$  mede o desvio padrão máximo da partilha de recursos. Por sua vez,  $W_s$  é uma janela de tempo durante a qual a média é calculado e onde o compartilhamento de recursos mais o desvio deve ser garantido. Por fim,  $D_s$  especifica a duração do tempo de vida do *slice*. O treinamento da rede neural se deu com a utilização de dados provenientes de 24h de execução de simulação de tráfego no *NS-3 Network Simulator*<sup>1</sup>. Ao lamentar o desempenho alcançado no treinamento da rede, Autor conclui sobre a necessidade de gerar mais dados para fornecer à rede LSTM.

Em abordagem do contexto de *slice* para redes veiculares (V2X), uma variação da arquitetura LSTM em conjunto com as Redes Neurais Convolucionais (CNN) é explorada por Cui [14]. O treinamento da rede neural utilizou um dataset de tráfego de rede móveis da cidade de Milão, na Itália [15], o qual contém dados de três categorias: SMS, telefone e navegação na web. Para efeito da simulação considerou cada categoria um *slice* a ser alocado através do modelo que, por sua vez, apresentou desempenho satisfatório em estabelecer a ligação entre o cliente e o *slice* que o atenderia. A rede LSTM é do tipo *Encoder/Decoder*, sendo que no projeto em questão, ocorreu a utilização do codificador para de prever o tráfego da rede, enquanto o decodificador possuía a atribuição de realizar a obtenção do *slice* ótimo. Todo o conjunto de dados do dataset é utilizado para o treinamento da rede, possuindo sua entrada na rede como uma matriz de uma imagem. Apesar do relato de resultado satisfatório, não é apresentado o SLA, nem a parametrização dos atributos do dataset utilizados, desta forma, replicar o procedimento para validar o desempenho da rede não é viável.

### III. ANÁLISE E VISUALIZAÇÃO DE DADOS

A utilização de metodologias estabelecidas no universo de aprendizado de máquina [6], pode vir a contribuir e instituir um método factível e replicável para o desenvolvimento de modelos para diferentes cenários de redes de computadores. O método, resumidamente, é estruturado em três fases:

- Pré-processamento: seleção de dados, limpeza, normalização;
- Mineração de Dados: processamento por algoritmo estatístico ou IA;
- Pós-processamento: realização de análise e validação de padrões.

Desta forma, a Descoberta de Conhecimento em Base de Dados permite a reflexão sobre a utilização de diversos modelos de algoritmo na fase de mineração. Esta abordagem pode gerar a padronização de dados de entrada provenientes de

Tabela I  
ESPECIFICAÇÃO DE LARGURA DE BANDA PARA CADA USER EQUIPMENT NOS TESTES REALIZADOS.

Teste	UE 01 (Mbps)	UE 02 (Mbps)	UE 03 (Mbps)	UE 04 (Mbps)	UE 05 (Mbps)
01	20	47	07	09	06
02	22	45	06	12	08
03	24	49	09	10	06
04	32	55	14	18	12
05	25	50	10	05	02
06	45	70	30	05	03
07	12	25	05	20	30
08	20	22	24	02	07
09	08	28	05	10	03
10	32	62	22	15	08
11	10	15	05	02	15

tráfegos de redes reais ou de simulações. Independentemente do modelo a ser utilizado, a coleta e geração de dados deve prover uma assinatura mínima deste dados, para garantir a diversidade na aplicação de algoritmos na fase dois do processo de análise.

Com a pretensão de avaliar a metodologia no contexto de tráfego de redes para a análise associada a utilização de *slice*, optou-se por definir um cenário controlado e gerar os dados com o atendimento a uma assinatura mínima e com baixo custo de aquisição.

#### A. Cenário dos Experimentos

Dada a necessidade de um dataset de tráfego de redes para a exploração, validação e desenvolvimento de modelos de ML, este trabalho optou por seguir o comportamento padrão apresentado em diversos artigos, o de construindo um cenário próprio e produzindo um dataset proveniente de uma simulação. Vale ressaltar ser compreensível este comportamento devido as característica amplas e diversas de cenários e necessidades das redes de computadores.

O ambiente de testes foi implementado utilizando o emulador de software de rede GNS3 (Graphical Network Simulator 3), com o objetivo de disponibilizar a infraestrutura para as funções virtualizadas de rede (NFV Infrastructure). O cenário é composto de 05 clientes transmitindo em diferentes taxas de tráfego, conforme especificado na Tabela I, para o conjunto de testes realizados. Além disso, tem-se a implementação na estação rádio base 5G (gNodeB) através de um módulo de software (CU/DU) executando em uma VM Ubuntu Linux 16.04 LTS. No que tange, a borda da rede (Edge Computing) utilizou-se um container docker com o deploy do controlador SDN OpenDayLight, para implementar a função da UPF (User Plane Function), responsável pela separação e identificação dos fluxos de dados. Além disso, fez-se a implementação da função de rede NWDAF (Network Data Analytics Function) em uma VM Ubuntu Linux 16.04 LTS com o framework python Anaconda 3, com o objetivo de implementar as funções de análise e o pipeline de mineração de dados, fazendo uso da ferramenta jupyter notebook e das bibliotecas Numpy, Pandas e Sklearn.

<sup>1</sup><http://www.nsnam.org/>

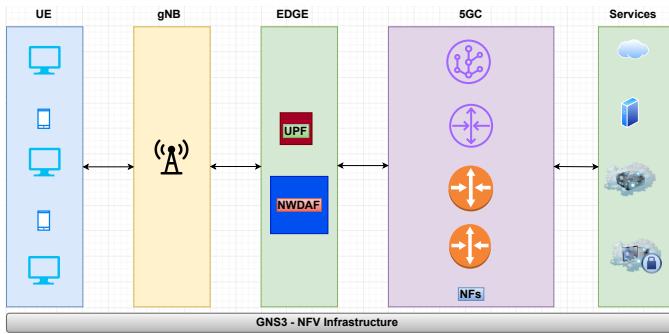


Figura 1. Ilustração da arquitetura do ambiente de simulação para produção do tráfego de rede.

É importante destacar que embora o cenário apresente o núcleo da arquitetura 5G (5G Core – 5GC), não implementou-se nenhuma das funções de rede (NFs) pertinentes ao core, tais como: AUSF, NEF, NRF, PCF, NSSF e UDM, conforme apresentado na Figura 1. Essas funções de rede serão implementadas através do Open Source Mano (OSM) da ETSI NFV, que fará uso do template SLA gerado pela função NWDAF. Contudo, considerou-se 03 slices verticais, detalhados a seguir, entregues na borda da rede pelo operador 5G. Em relação a representação dos serviços (services) utilizou-se um servidor Iperf, para diferentes taxas de transmissão de tráfego do tipo CBR (Constant Bit Rate).

- Slice 01: Remote Driving
  - E2E Latency (5ms) → máxima
  - Reability (99.999%) → mínima
  - Data rate
    - \* (UL: 25Mbps) → mínima
    - \* (DL: 1 Mbps) → mínima
- Slice 02: Rural Macro
  - E2E Latency (Não especifica)
  - Reability (Superior a 80%)
  - Data rate
    - \* (UL: 50Mbps)
    - \* (DL: 25Mbps)
- Slice 03: Wireless Road-Side Infrastructure Backhaul (ITS)
  - E2E Latency (30ms) → máxima
  - Reability (99.999%)
  - Data rate: 10Mbps

Como resultado para cada um dos onze experimentos da simulação, obteve-se um arquivo para cada cliente contendo 33 dados de conexões com intervalo, transferência, banda e protocolo. Para os dados de avaliação de cada slice, para cada experimento foi gerado um arquivo contendo latência, jitter, perda de todas as conexões realizadas. Desta forma, estes dados brutos de log são o ponto de partida para a realização do processo de análise de dados.

### B. Descoberta de Conhecimento

No processo tradicional de análise de dados enamorar-se pelo conjunto de dados faz parte da primeira etapa. Neste

sentido, para conhecer a preparar os dados para o processo de mineração, seguiu-se com as tarefas do pré-processamento de dados:

- Coleta de dados da simulação;
- Seleção de dados:
- Purificação de dados;
- Construção de dataset;

Para efeito de experimento foram considerados seis 6 atributos a serem organizados no dataset, contudo o operador da rede pode adicionar métricas para atender cenários diversos desde que possa coletar os dados da rede. Para validar este princípio, foram adicionados outros dois critérios ao dataset: confiabilidade e distância. Estes, por sua vez, foram gerados aleatoriamente respeitando um intervalo pré-definido dentro do normalmente existente em cenários reais. Assim, o dataset padrão passou a possuir os seguintes atributos:

- Latency
- Jitter
- Loss
- Protocol
- Bandwidth
- Transfer
- UE: User Equipment, não utilizado no processo de construção do modelo.
- Experiment: id do experimento, não utilizado no processo de mineração.
- Distance

No pré-processamento os dados são analisados quanto ao tipo, intervalo e, nesta fase é comum a ocorrência de ações de tratamento e normalização aplicados aos registros. Desta maneira, os dados possam vir a servir e adequar-se ao padrão de entrada do algoritmo de mineração de dados utilizado na etapa futura. No sentido de reconhecer os dados, para efeito de ilustração, as Figuras 2 e 3 apresentam os gráficos de distribuição de densidade dos atributos latência e perda.

No processo de mineração houve uso do algoritmo de classificação ou agrupamento K-Means que utiliza dados numéricos. Por este motivo, o atributo Protocolo sofre a transformação de dados, de maneira que passa a ser '0' para 'TCP' e '1' para 'UDP'. Para o processo de aprendizagem passou a considerar somente os atributos relevantes para o experimento:

['Latency', 'Jitter', 'Loss', 'Protocol', 'Bandwidth', 'Transfer', 'Distance', 'Reliability']

O agrupamento realiza a análise de dados em busca de reconhecer o comportamento de grupo dos dados de maneira que um elemento seja caracterizado em um grupo e claramente seja diferente dos outros grupos. O k do K-Means indica quantos são os grupos separáveis do conjunto de dado. Uma técnica para encontrar o valor adequado de k para um dado conjunto é aplicar o Método de Elbow Plot pelo cálculo da soma do erro médio quadrático (SSE). O ponto de Elbow ocorre no momento que o SSE começa a diminuir de forma linear. Desta maneira, no dataset utilizado, o k apropriado é

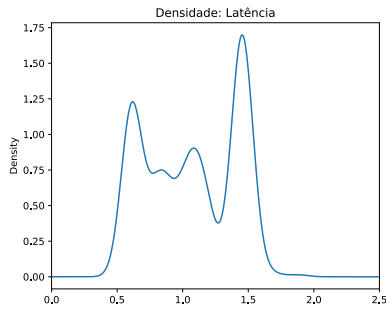


Figura 2. Distribuição de densidade da latência.

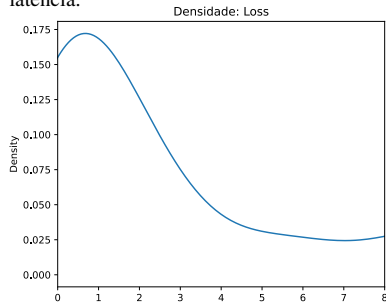


Figura 3. Distribuição de densidade da perda.

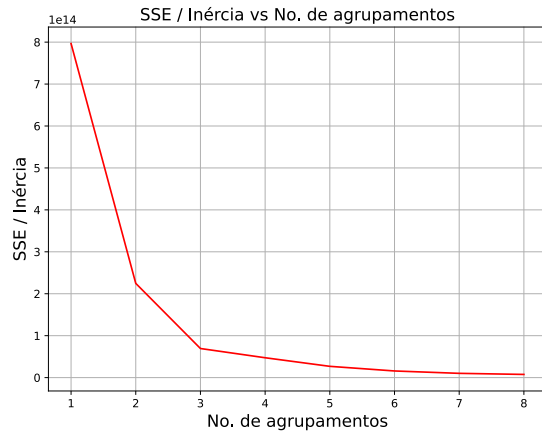


Figura 4. Descida do Erro Médio Quadrático para K = 1 até K = 8.

3, conforme pode ser observado na Figura 4 que apresenta o SSE para k de 1 a 8.

Por fim, ao assumindo o k igual a 3 ocorre o processo de aprendizado e treinamento com a separação dos elementos em três grupos que caracterizam cada slice. A etapa seguinte à mineração de dados é a análise onde busca-se entender o resultado encontrado. Para esclarecer esta etapa a ilustração da Figura 5 apresenta as características únicas de cada agrupamento que, de certa forma, refletem as características de cada slice utilizado no sistema de simulação GNS3. Uma vez o modelo treinado, o sistema passa a ser capaz de receber qualquer fluxo de dados dentro do intervalo médio do treinamento, para forçar a predição do *slice* adequado para a alocação.

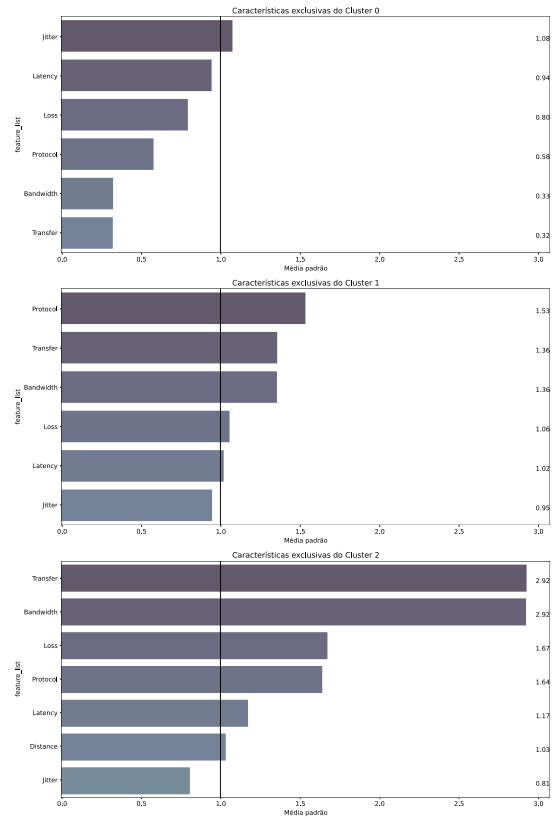


Figura 5. Caracterização única dos agrupamentos encontrados no processo.

Na etapa de análise do resultado produzido pelo processo de mineração é explorada a capacidade de gerar ilustrações, sempre no sentido de validar o conhecimento adquirido. Tal conhecimento deve ser útil, novo e aplicável ao contexto do problema.

#### IV. RESULTADOS E CONTRIBUIÇÕES

Para efeito de experimento, o processamento realizado mostrou-se viável, no sentido de formalizar um dataset proveniente da própria rede, sendo este não intrusivo quanto a sensibilidade dos dados da rede. Também mostrou-se versátil, pois o procedimento permite substituir o algoritmo de K-Means, apropriado para agrupamento, por outros em busca de outras tarefas do processo de análise de dados. A possibilidade de explorar algoritmos de aprendizado por reforço, ou ainda, as redes neurais recorrentes, como a LSTM, exigem apenas que o processo de coleta, a partir de simulação ou um ambiente real, ocorra por mais tempo para produzir um dataset maior. Assim, sua assinatura permanece a mesma, mas permitindo que novos atributos sejam adicionados ao arquivo.

Apesar de, por enquanto, a análise do fluxo desconsiderar a abordagem fim-a-fim, ocorre a otimização do tráfego da rede de acesso à borda. Desta forma, de posse do modelo de agrupamento de dados e capacidade de separar o *slice* adequado ao usuário considerando seu contexto de fluxo, torna-se possível iniciar o processo de classificação para recomendar o *slice* ao usuário. Dada a assinatura do fluxo de dados exigido

para o modelo e, por fim, sua capacidade de recomendar um *slice* considerando suas características, como trabalhos futuros busca-se utilizar outros modelos de aprendizagem de máquina na etapa de mineração de dados, além de explorar algoritmos para transcrever as características do *slice* em formato exigido pelas ferramentas de orquestração, desta forma permitindo que o processo de redes automatizada alcance mais um degrau.

#### REFERÊNCIAS

- [1] M. C. Soveri, "Management of network slicing in mobile networks; concepts, usecases and requirements, 3gpp ts 28.530, v0.5.0," techreport, Feb. 2018.
- [2] K. Kymalainen, "5G System; Network Slice Selection Services; Stage 3, 3GPP TS 29.531, v1.0.0," 3GPP Organizational Partners, techreport, Mar. 2018.
- [3] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, "On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control," *IEEE Network*, vol. 33, pp. 242–252, 2019.
- [4] X. Vasilakos, N. Nikaen, D. H. Lorenz, B. Koksai, and N. Ferdosian, "Integrated methodology to cognitive network slice management in virtualized 5g networks," May 2020.
- [5] W. Guan, H. Zhang, and V. C. M. Leung, "Analysis of traffic performance on network slicing using complex network theory," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 188–15 199, 2020.
- [6] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, p. 37, Mar. 1996. [Online]. Available: <https://ojs.aaai.org/index.php/aimagazine/article/view/1230>
- [7] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5g network slicing using sdn and nfv- a survey of taxonomy, architectures and future challenges," 2019, Dec. 2019.
- [8] L. Li, K. Ota, and M. Dong, "DeepNFV: A Lightweight Framework for Intelligent Edge Network Functions Virtualization," *IEEE Network*, vol. 33, no. 1, pp. 136–141, 2019.
- [9] Q. Liu, T. Han, and E. Moges, "EdgeSlice: Slicing Wireless Edge Computing Network with Decentralized Deep Reinforcement Learning," Mar. 2020.
- [10] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Reinforcement learning for dynamic resource optimization in 5g radio access network slicing," Sep. 2020.
- [11] M. Usama, R. N. Mitra, I. Ilahi, J. Qadir, and M. K. Marina, "Examining Machine Learning for 5G and Beyond through an Adversarial Lens," Sep. 2020.
- [12] R. Wen, G. Feng, J. Tang, T. Q. S. Quek, G. Wang, W. Tan, and S. Qin, "On robustness of network slicing for next-generation mobile networks," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 430–444, 2019.
- [13] M. Toscano, F. Grunwald, M. Richart, J. Baliosian, E. Grampín, and A. Castro, "Machine learning aided network slicing," in *2019 21st International Conference on Transparent Optical Networks (ICTON)*, 2019, pp. 1–4.
- [14] Y. Cui, X. Huang, D. Wu, and H. Zheng, "Machine learning based resource allocation strategy for network slicing in vehicular networks," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, 2020, pp. 454–459.
- [15] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of milan and the province of trentino," *Scientific Data*, vol. 2, no. 1, p. 150055, Oct 2015. [Online]. Available: <https://doi.org/10.1038/sdata.2015.55>