

# Avaliação do uso de Aprendizado de Máquina para Engenharia de Tráfego em Redes Cientes de Caminho

Beatriz Mariano<sup>1</sup>, Cristina Dominicini<sup>1</sup>, Domingos Paraíso<sup>1</sup>,  
Eduarda Coelho<sup>1</sup>, Daniel Ventorim<sup>2</sup>, Giovanni Comarela<sup>2</sup>, Magnos Martinello<sup>2</sup>

<sup>1</sup>Instituto Federal do Espírito Santo (IFES)

<sup>2</sup>Universidade Federal do Espírito Santo (UFES)

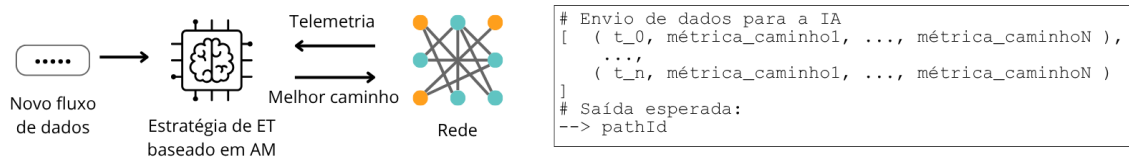
beatriz-auer@hotmail.com, cristina.dominicini@ifes.edu.br

**Abstract.** *This paper investigates the application of supervised machine learning to predict the lowest latency path in Path-Aware Networks. Initially, the analysis demonstrates that public datasets from traditional networks are unsuitable due to structural limitations and the lack of guarantees regarding traversed paths. To bridge this gap, an emulated prototype based on Software-Defined Networking (Mininet and Ryu) was developed to generate precise telemetry datasets under diverse traffic profiles. Although classifiers exceed 97% accuracy in previously known scenarios, the evaluation reveals they tend to memorize the predominant class from the training set rather than assimilating the implicit lowest latency rule. These findings highlight the need for continuous retraining mechanisms in dynamic environments and establish the proposed testbed as a valuable tool for the empirical validation of future intelligent routing solutions.*

**Resumo.** *Este artigo investiga a aplicação de aprendizado de máquina supervisionado para prever o caminho de menor latência em Redes Cientes de Caminho. Inicialmente, a análise demonstra que datasets públicos de redes tradicionais são inadequados devido a limitações estruturais e à ausência de garantias sobre os trajetos percorridos. Para suprir essa lacuna, desenvolveu-se um protótipo emulado baseado em Redes Definidas por Software (Mininet e Ryu), capaz de gerar datasets precisos de telemetria sob diversos perfis de tráfego. Embora os classificadores superem 97% de acurácia em cenários previamente conhecidos, a avaliação revela que eles tendem a memorizar a classe predominante do treinamento em vez de assimilar a regra implícita da menor latência. Esses achados evidenciam a necessidade de retreinamento contínuo em ambientes dinâmicos e estabelecem o ambiente proposto como uma ferramenta valiosa para a validação empírica de futuras soluções de roteamento inteligente.*

## 1. Introdução

O crescimento anual do tráfego global de dados, impulsionado pela expansão da Internet das Coisas e pelo consumo de conteúdos multimídia pela Internet [Sandvine 2024], impõe desafios à gestão da infraestrutura de rede [International Telecommunication Union 2023, Sandvine 2024]. O volume de dados trafegados já alcança a escala de exabytes, reforçando a necessidade de um gerenciamento de rede mais eficiente. Nesse contexto, um dos principais desafios da Engenharia de Tráfego (ET), área de estudo responsável por definir políticas de encaminhamento e uso eficiente dos recursos, é desenvolver



**Figura 1. Proposta para aplicação de AM para a escolha de caminhos em redes**

mecanismos de controle capazes de se adaptar rapidamente às variações do estado da rede sem comprometer sua estabilidade [Farrel 2024]. No entanto, métodos tradicionais de ET, baseados em heurísticas estáticas e em uma visão global da rede, apresentam limitações na resposta a mudanças dinâmicas, além de elevada complexidade computacional [Farrel 2024, Zhou et al. 2025].

Na arquitetura tradicional da Internet a rede é tratada como uma infraestrutura opaca, na qual apenas a conectividade entre *endpoints* importa, enquanto o caminho percorrido permanece invisível para as aplicações [Trammell 2022]. Em resposta, Redes Cientes de Caminho buscam permitir que *endpoints* influenciem a seleção de rotas conforme os requisitos de cada fluxo [Trammell 2022]. Nesse contexto, a telemetria de rede torna-se fundamental para o desenvolvimento de controladores baseados em Aprendizado de Máquina (AM), capazes de identificar padrões e detectar congestionamentos e falhas a partir da análise das métricas coletadas [Yu 2019, Al-Najjar et al. 2024]. Estudos recentes demonstram que o campo da AM tem amadurecido para ser aplicado em cenários reais de ET, apresentando desempenho equiparado ao estado da arte em ET tradicional [Bernárdez et al. 2021].

Contudo, a aplicação prática de AM na ET enfrenta desafios como a escassez de *datasets* e a falta de validação em redes reais. Embora muitos trabalhos utilizem topologias reais em seus experimentos, ainda é comum o uso de tráfego sintético, devido à indisponibilidade de dados públicos e a restrições impostas por provedores [Ferreira et al. 2023]. Além disso, grande parte dos modelos é avaliada apenas em ambientes controlados, com base em métricas de erro, o que dificulta aferir sua eficácia em cenários reais [Ferreira et al. 2023]. Como consequência, os resultados podem não representar a otimização real da rede quando aplicados em cenários realistas.

Diante deste cenário, este artigo investiga a aplicação de algoritmos de aprendizado de máquina supervisionado para a predição do melhor caminho em redes cientes de caminho, utilizando séries históricas de telemetria. A Figura 1 ilustra a arquitetura funcional proposta: ao ser consultado, o mecanismo de ET baseado em AM recebe os dados históricos mais recentes de telemetria da rede, realiza a análise comparativa entre os caminhos disponíveis e recomenda a rota mais adequada.

A tarefa de seleção de rota é formulada como um problema de comparação entre as métricas dos caminhos ao longo do tempo. Para cada instante  $t$ , são comparados os valores coletados em todos os caminhos disponíveis entre os *endpoints*. Como mostra a Figura 1, o algoritmo de AM deve produzir como saída o identificador da rota considerada mais apropriada, representado pelo valor  $pathId \in (\mathbb{Z}_+^*)$ , número entre 1 e  $n$ , sendo  $n$  a quantidade de caminhos disponíveis. Para delimitar o escopo desta análise, o critério adotado neste estudo para a definição da melhor rota é a minimização da latência. Contudo, a modelagem proposta permite futuras extensões para incorporar o uso combinado

de outras métricas de rede, como *jitter*, perda de pacotes e vazão.

Assim, considerando as limitações das abordagens tradicionais de ET e o potencial das redes cientes de caminho, este trabalho é guiado pelas seguintes questões: (1) os *datasets* de medições de rede existentes permitem extrair métricas por caminho na rede? (2) como construir *datasets* de telemetria adequados a redes cientes de caminho? e (3) como utilizar esses dados para treinar e validar modelos de AM na seleção de caminhos?

Para enfrentar esses desafios, as principais contribuições deste trabalho para redes cientes de caminho são: (i) **análise da aplicabilidade de *datasets* públicos coletados em ambientes tradicionais**: o estudo evidencia que fatores estruturais, como o forte desbalanceamento e a ausência de identificação precisa dos trajetos das sondas, inviabilizam o uso desses *datasets* para redes cientes de caminho; (ii) **construção de um ambiente emulado para geração de *datasets***: desenvolvimento de um protótipo SDN com Mininet e Ryu, capaz de instanciar topologias e coletar métricas, como latência, para criação de *datasets* cientes de caminhos sob diferentes cenários de tráfego; (iii) **avaliação de métodos supervisionados de AM para determinar o caminho de menor latência**: treinamento e comparação do desempenho de classificadores clássicos, sob diferentes cenários.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 discute a aplicabilidade de *datasets* de redes tradicionais; a construção do ambiente emulado de geração de telemetria; e os *datasets* produzidos. A Seção 4 discute a avaliação experimental da generalização dos modelos de aprendizado de máquina. Por fim, a Seção 5 traz as conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

Historicamente, a ET baseou-se em estruturas matemáticas como teoria das filas, simulações de *Monte Carlo* e técnicas estatísticas, assumindo uma visão global e usando heurísticas [Kiran et al. 2022]. Nos protocolos tradicionais de roteamento, como o OSPF (Open Shortest Path First), as decisões de roteamento apoiam-se em métricas estáticas, como o custo do enlace (geralmente associado à largura de banda). Para otimizar a vazão, o mecanismo ECMP (Equal-Cost Multi-Path) distribui o tráfego entre múltiplos trajetos de custo equivalente. No entanto, por limitar-se estritamente às rotas de mesmo custo, essa estratégia frequentemente resulta na subutilização dos recursos globais da rede [Yadav 2018].

O estado da arte em otimização clássica é representado por algoritmos refinados como o DEFO (*Declarative and Expressive Forwarding Optimizer*), que utiliza Programação de Restrições como seu método de seleção de caminhos, um paradigma de otimização onde os objetivos da rede, como minimizar a carga dos enlaces ou respeitar limites de atraso, são modelados como variáveis matemáticas sujeitas a restrições que devem ser satisfeitas simultaneamente [Hartert et al. 2015]. O DEFO opera de forma centralizada e consegue otimizar redes em minutos [Bernárdez et al. 2021].

A dificuldade em gerenciar a complexidade e a variabilidade do tráfego têm impulsionado a exploração de métodos de AM para o desenvolvimento de soluções otimizadas e preditivas [Al-Najjar et al. 2024]. O paradigma *Knowledge-Defined Networking* (KDN) integra Redes Definidas por Software (SDN), *Network Analytics* (NA) e AM para transformar dados de telemetria em conhecimento acionável, automatizando a tomada de decisões [Mestres et al. 2017].

Na ET, o AM é geralmente aplicado em três categorias: aprendizado supervisionado para prever o desempenho da rede; aprendizado não supervisionado para descobrir correlações e estruturas de dados; e aprendizado por reforço (RL/DRL) para definir ações que maximizem uma recompensa [Aleksseva et al. 2021]. Um exemplo é o **Hecate**, que utiliza GNN para aprender perfis de tráfego e prever estatísticas futuras da rede e, em seguida, aplica DRL ou busca gulosa para otimizar o roteamento com múltiplos objetivos [Kiran et al. 2022]. Outra abordagem é o **MARL+GNN**, que combina *Multi-Agent Reinforcement Learning* e GNN em uma arquitetura distribuída, reduzindo o tempo de execução de minutos para segundos com desempenho comparável ao DEFO [Bernárdez et al. 2021].

Apesar desses avanços, grande parte dos modelos propostos de AM ainda são avaliados apenas em ambientes controlados, limitando-se a comparações com métricas de erro, o que limita a verificação de sua eficácia em redes reais [Ferreira et al. 2023]. Além disso, a escassez de conjuntos de dados reais e públicos e as restrições de acesso impostas por provedores fazem com que diversos estudos dependam de topologias simuladas ou dados sintéticos.

A Tabela 1 compara as principais soluções de ET discutidas. Protocolos tradicionais, como OSPF e ECMP, apresentam limitações significativas em ambientes dinâmicos devido à natureza opaca da rede e ao uso de métricas rígidas [Kurose and Ross 2017, Trammell 2022]. Embora abordagens de ET baseadas em AM ofereçam otimizações avançadas, não há publicações de validação desses modelos em cenários realistas. As contribuições fundamentais da nossa proposta consistem na demonstração da inadequação de *datasets* públicos coletados em ambientes tradicionais para o contexto de redes cientes de caminho [Paullada et al. 2021] e no desenvolvimento de um ambiente experimental robusto para a geração de telemetria realista, facilitando a integração de modelos de AM à infraestrutura de rede [Al-Najjar et al. 2024, Paraíso 2023].

Nos protocolos tradicionais de roteamento, como o OSPF (Open Shortest Path First), as decisões de roteamento apoiam-se em métricas estáticas, como o custo do enlace (geralmente associado à largura de banda). Para otimizar a vazão, o mecanismo ECMP (Equal-Cost Multi-Path) distribui o tráfego entre múltiplos trajetos de custo equivalente. No entanto, por limitar-se estritamente às rotas de mesmo custo, essa estratégia frequentemente resulta na subutilização dos recursos globais da rede.

### 3. Projeto e Desenvolvimento do Ambiente Emulado

#### 3.1. Inadequação dos *datasets* existentes

A aplicação de algoritmos de AM na predição de rotas em redes cientes de caminho exige o uso de *datasets* capazes de representar cada rota existente entre dois *endpoints* de forma confiável. Para validar a aplicabilidade de bases de dados pré-existentes, este trabalho analisou inicialmente um *dataset* público de telemetria disponibilizado pela Rede Nacional de Ensino e Pesquisa (RNP) em 2024, coletado no contexto de um desafio de AM [Comitê Técnico de Monitoramento de Redes – RNP 2024]. O conjunto de dados contém medições de *traceroute* e *Round Trip Time* (RTT) em uma topologia formada por nós da rede acadêmica nacional. Apesar de representar uma infraestrutura real, a análise revelou que os dados coletados em redes tradicionais apresentam quatro limitações significativas para o contexto de redes cientes de caminho. São elas:

Tabela 1. Comparação de Trabalhos Relacionados em Engenharia de Tráfego

Método	AM	Ciência de caminho	Dataset realista	Método de seleção	Desvantagens
OSPF	×	×	×	Custo de enlace	Rigidez e métricas estáticas
ECMP	×	×	×	Hashing estático	Ineficiente em redes assimétricas
DEFO	×	×	✓	PR e Busca em Grandes Vizinhanças	Complexidade de reconfiguração
MARL+GNN	✓	×	✓	MARL + GNN	Não validado em cenário realista
Hecate	✓	✓*	×	DRL + GNN	Não validado em cenário realista
Proposto	✓	✓	✓**	Aprendizagem Supervisionada	Depende da qualidade do conjunto de treino

\* O método Hecate utiliza apenas um dataset muito simplificado de ciência de caminhos, com medições da largura de banda em apenas dois enlaces Wi-Fi e LTE por 500 segundos [Al-Najjar et al. 2024].

\*\* Os datasets foram obtidos em ambiente emulado sob perfis de tráfego realistas.

- **Falta de garantia de caminho:** As medições de *traceroute* e RTT não garantem que os pacotes de teste percorreram caminhos fixos e determinísticos, o que torna a latência reportada pouco confiável para representar um caminho específico.
- **Desalinhamento temporal:** As coletas de diferentes caminhos entre uma mesma origem e destino foram realizadas em instantes (*timestamps*) distintos, inviabilizando a comparação simultânea do estado da rede.
- **Frequência irregular:** O espaçamento temporal entre as amostras variou drasticamente; enquanto algumas medidas apresentaram intervalos de minutos, em outras havia dias de distância, o que prejudica a análise de séries temporais e exige interpolações que podem introduzir vieses artificiais.
- **Forte desbalanceamento de classes:** Constatou-se que aproximadamente 88% das medições representavam um único caminho dominante. Esse cenário degenerado induz os classificadores a um viés trivial, fazendo com que os modelos apenas memorizem a classe majoritária sem aprender regras de generalização.

No contexto de redes cientes de caminho, o desbalanceamento observado neste *dataset* soma-se as outras limitações. O problema não é a predominância de um caminho frequentemente ótimo, mas a baixa representatividade das demais rotas e visão global do estado da rede. Como esses caminhos são pouco observados, o modelo não tem visibilidade sobre seu comportamento, o que compromete sua capacidade de generalização. Como consequência, tende a reproduzir o comportamento do roteamento tradicional, reforçando o caminho dominante pela falta de evidências sobre alternativas. Essa limitação é crítica, pois a decisão depende da comparação entre múltiplas rotas.

Devido a essas limitações, foi observado que o uso de *datasets* coletados em redes tradicionais costumam ser inadequados para treinar AM no contexto de redes cientes de caminho. Diante desse problema, projetou-se um protótipo emulado capaz de gerar conjuntos de dados específicos para redes cientes de caminho.

### 3.2. Arquitetura do Protótipo

Com o objetivo de criar novos *datasets* aplicáveis ao contexto de redes cientes de caminho, foi desenvolvido um protótipo emulado baseado em SDN utilizando o emulador Mininet [Contributors 2025] integrado a um controlador Ryu [Kubo et al. 2014]. Esse arranjo permite configurar topologias, rotas e agentes de telemetria de forma simples e automatizada. Além disso, o protótipo foi construído de modo a permitir a integração e validação de diferentes métodos de ET, como tomadores de decisão baseados em AM. A arquitetura proposta é composta por quatro módulos independentes que interagem entre si: Testes e Telemetria, Rede, Controlador e Gerenciamento de Tráfego. A Figura 2 apresenta o protótipo e seus componentes, conforme descrição a seguir.

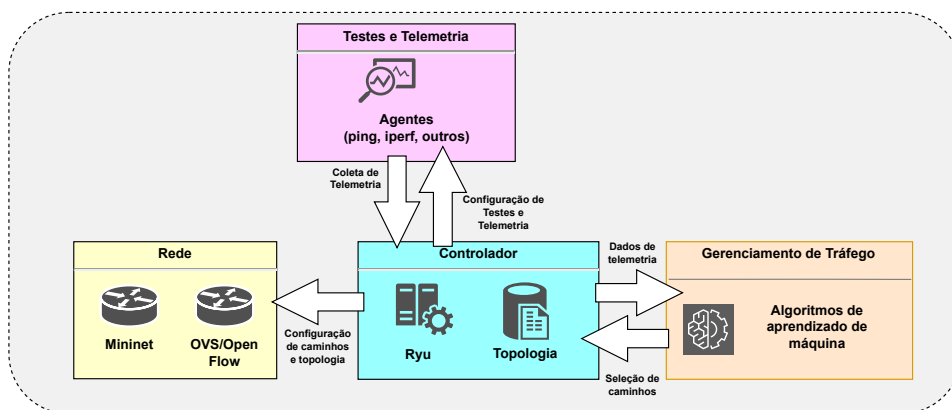


Figura 2. Arquitetura proposta para o protótipo

1. **Módulo de Testes e Telemetria:** Esse módulo é responsável por criar dois tipos de agentes para injeção de tráfego ao longo da execução dos experimentos: sondas de monitoramento contínuo através de coleta de telemetria de rede, compostas por pacotes ICMP que seguem caminhos específicos configurados por regras OpenFlow [McKeown et al. 2008] pré-instaladas; e o tráfego de testes, baseado em fluxos TCP e UDP, que simulam a carga de trabalho da rede. Enquanto as sondas possuem caminhos fixos para garantir que a medição corresponda a um trajeto específico, o tráfego de carga pode seguir estratégias de roteamento dinâmico, como ECMP ou OSPF, ou fixar um caminho específico com regras instaladas dinamicamente pelo controlador no momento em que os fluxos são iniciados.
2. **Módulo de Rede:** Responsável pela infraestrutura experimental. O Mininet emula a topologia lógica, os *hosts* e os enlaces, enquanto o Open vSwitch (OVS) atua nos dispositivos para implementar o controle de fluxos via protocolo OpenFlow. Permite configurar latência, largura de banda e perda de pacotes dos enlaces.
3. **Módulo Controlador:** Implementado com o *framework* Ryu, atua como o elemento central de tomada de decisão. Detém a visão global da topologia, recebe os dados de telemetria, processa eventos e instala as regras de encaminhamento dinâmicas nos *switches*. Além disso, mantém históricos temporais de telemetria interpolados em uma base temporal comum.
4. **Módulo de Gerenciamento de Tráfego:** Analisa métricas em tempo real e históricas por meio de algoritmos de otimização e AM. Identifica congestionamentos, prevê variações e determina rotas adequadas para cada fluxo. As decisões

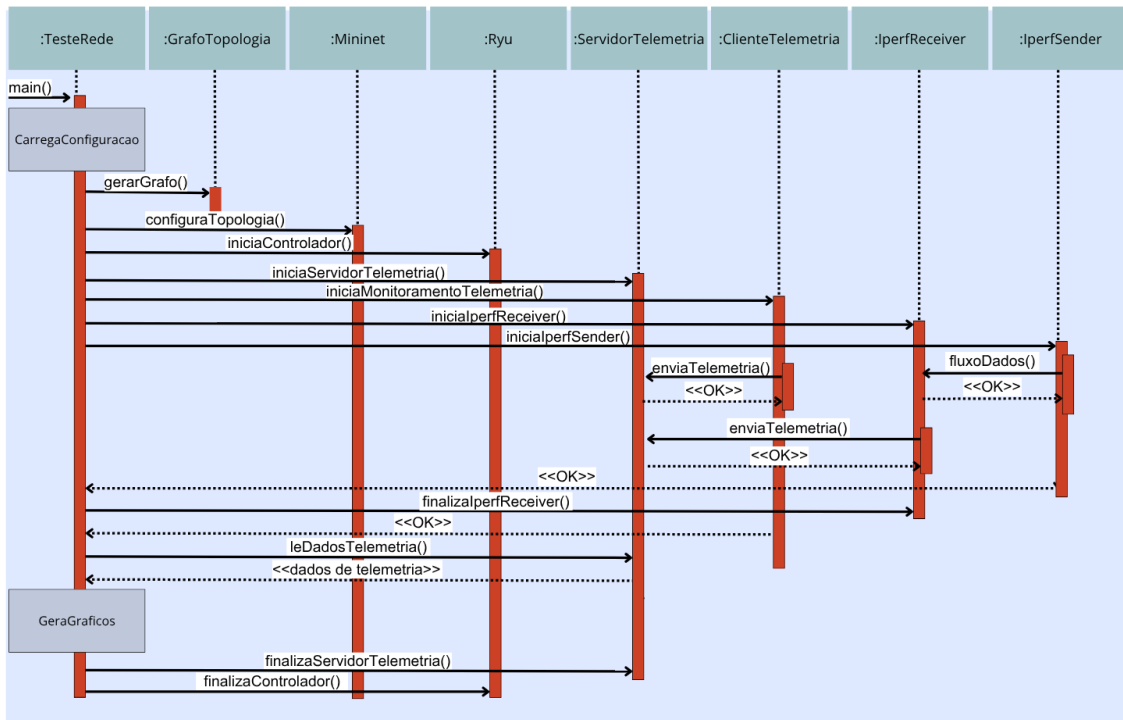


Figura 3. Diagrama de sequência do protótipo desenvolvido

são transferidas ao Controlador, que as aplica na topologia. Esse módulo é externo ao protótipo e interage com ele a partir de interfaces de aplicação.

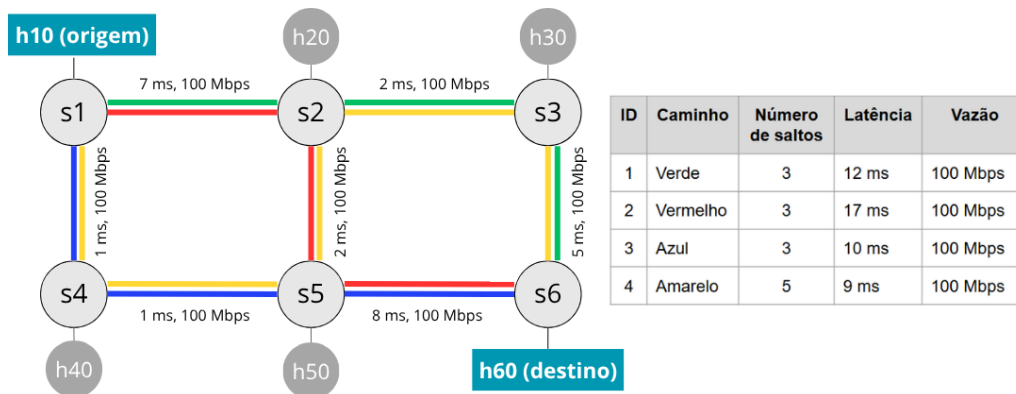
O protótipo segue um fluxo operacional que integra todos os módulos da arquitetura, desde a preparação do ambiente até a geração dos relatórios. Como ilustrado na Figura 3, a execução inicia-se com o carregamento das configurações do experimento. Para garantir a flexibilidade das experimentações, o protótipo é integralmente configurado por meio de um arquivo no formato JSON. Nesse arquivo, é possível definir o método de roteamento utilizado, a topologia da rede, o tipo de telemetria a ser coletada e os parâmetros dos itens de teste responsáveis por introduzir carga na rede. Essa estrutura de configuração permite emular virtualmente qualquer topologia e cenário dinâmico de tráfego.

Após essa etapa, o Mininet instancia os *hosts*, *switches* OpenFlow e enlaces com seus respectivos atributos. Então, o controlador assume o gerenciamento central, processando eventos e instalando regras de encaminhamento. Com a infraestrutura estável, o servidor de telemetria é ativado e os agentes distribuídos iniciam medições periódicas de latência, definida como a métrica de RTT de requisições com a ferramenta *ping*, e vazão, monitorada através da ferramenta *bwm-ng*. As medições são enviadas continuamente ao servidor. Cada agente de telemetria é designado exclusivamente a cada rota possível entre dois *hosts* da topologia. Para garantir o isolamento da leitura, cada rota recebe um identificador (*pathId*), definido automaticamente na inicialização do experimento.

Paralelamente aos fluxos de telemetria, o protótipo inicia os fluxos de testes dos experimentos utilizando a ferramenta *iperf*. Dessa forma, é possível gerar tráfego para representar a carga da rede, conforme os parâmetros de duração, intensidade e protocolos definidos na configuração. Enquanto o tráfego está ativo, o sistema mantém a coleta contínua de dados, permitindo avaliar congestionamento, variações de desempenho e o

efeito das estratégias de roteamento aplicadas.

Ao final dos experimentos, os fluxos e os serviços de telemetria são encerrados e as medições registradas são consolidadas em relatórios e gráficos para análise, abrangendo métricas, eventos e informações sobre as rotas utilizadas. Por fim, o ambiente é encerrado de forma ordenada, com a finalização do controlador, o encerramento dos processos auxiliares e a remoção da topologia do Mininet. Esse fluxo automatizado garante a reprodutibilidade dos experimentos e consolida o protótipo como uma plataforma integrada para estudos de engenharia de tráfego em redes cientes de caminho. O código-fonte do protótipo, bem como os *datasets* produzidos nesse artigo, foram disponibilizados em <https://github.com/auerbeatriz/path-aware-datasets>.



**Figura 4. Topologia e configurações do protótipo usadas no experimento para a criação de um *dataset*. Os quatro caminhos distintos entre  $h_{10}$  e  $h_{60}$  são monitorados por telemetria. Os demais hosts geram tráfego de fundo.**

Além de viabilizar a coleta de *datasets* realistas, o testbed proposto introduz uma infraestrutura de treinamento de AM em um ciclo de controle fechado e *online*. A integração entre os módulos de controle e gerenciamento de tráfego permite aplicar as decisões dos modelos em tempo real, possibilitando a avaliação precisa de sua eficácia na otimização da rede. O ambiente também suporta o treinamento *online* de agentes mais complexos, como DRL e GNNs, favorecendo a aprendizagem de políticas de ET com capacidade de generalização para diferentes topologias.

### 3.3. Metodologia de geração dos *datasets*

Para a geração dos *datasets* avaliados neste trabalho, o arquivo JSON do protótipo foi configurado para instanciar a topologia de quatro caminhos alternativos entre os *hosts* de origem ( $h_{10}$ ) e destino ( $h_{60}$ ), conforme Figura 4. Outros quatro *hosts* foram instanciados, cada um diretamente conectado aos demais *switches* da topologia para geração de tráfego de fundo, representando disputa por recursos. A partir disso, a rede foi submetida a diferentes condições de carga, resultando na geração de quatro *datasets* distintos, cada um correspondente a uma hora de coleta contínua:

- **D1: Rede Vazia:** Cenário contendo apenas sondas de telemetria, representando o estado das latências da rede sem nenhum tráfego de dados.
- **D2: Tráfego de fundo:** Introdução de fluxos TCP de 10 Mbps seguindo uma distribuição de Poisson, com intervalos médios de aproximadamente 1 minuto, saindo de todos os *hosts* com destino a todos os outros.

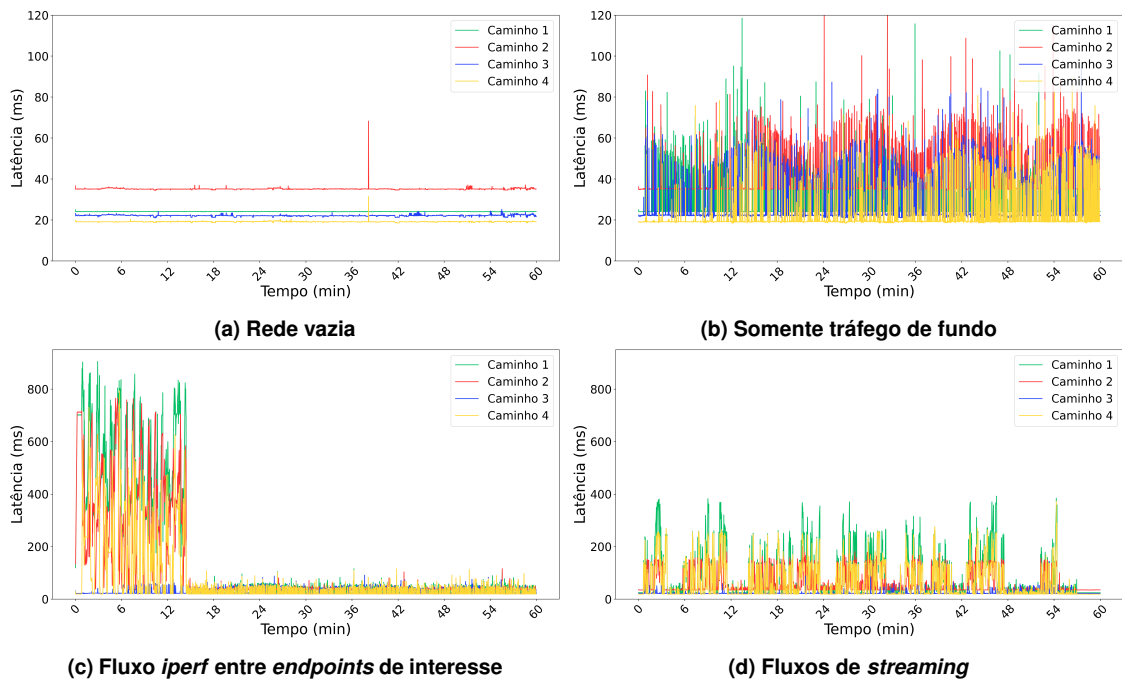


Figura 5. Latência por caminho nos *datasets* gerados no protótipo emulado.

- **D3: Fluxo TCP Contínuo:** Execução de uma transferência ininterrupta via *iperf* entre os *hosts* de origem e destino, ocorrendo de forma concorrente ao tráfego de fundo idêntico ao item anterior.
- **D4: Tráfego de Streaming:** Simulação do cenário apresentado no *dataset* da RNP [Comitê Técnico de Monitoramento de Redes – RNP 2024]. Durante o período de cada *sessão*,  $n$  fluxos *iperf* UDP são enviados a cada  $x$  segundos, do *host* destino para o de origem. Para adicionar *overhead* na rede, um ping é enviado da origem para o destino antes do início dos fluxos *iperf*. Os fluxos de *streaming* trafegam concorrentemente aos de tráfego de fundo. Três fluxos de *streaming* idênticos foram iniciados entre os *hosts* h10 e h60, em intervalos de 6 minutos, com 15 requisições a cada 3 minutos e intervalos de 6 minutos. Todos foram encerrados de forma sincronizada.

A análise dos gráficos de latência dos quatro *datasets* mostra que o Caminho 4 (Amarelo) mantém, de forma consistente, a menor latência ao longo do período de coleta, resultando em uma representação desbalanceada entre os caminhos. Nessas condições, espera-se que os modelos aprendam a prever o Caminho 4 como o de menor latência, independentemente de *outliers* causados por oscilações momentâneas da rede. Além disso, nas Figuras 5c e 5d, observa-se que o Caminho 3 (azul) também apresenta latência menor e estável, enquanto os Caminhos 1 e 2 (verde e vermelho) alternam entre si a menor latência. Para obter um conjunto de dados com maior alternância entre caminhos, os Caminhos 3 e 4 foram removidos, originando dois novos *datasets*:

- **D3a: Fluxo TCP Contínuo sem caminhos dominantes:** Subconjunto do dataset D3 com remoção dos Caminhos 3 e 4.
- **D4a: Tráfego de Streaming sem caminhos dominantes:** Subconjunto do dataset D4 com remoção dos Caminhos 3 e 4.

Como resultado, obtém-se uma distribuição de classes mais equilibrada, sem um caminho claramente dominante, o que permite explorar melhor a capacidade de generalização dos modelos de AM diante de mudanças no padrão de tráfego. Com o objetivo de diversificar ainda mais o conjunto de dados, foram gerados *datasets* adicionais para a mesma topologia, com modificações nas latências dos enlaces, alterando a ordem dos caminhos de menor para maior latência para Azul, Vermelho, Verde e Amarelo, respectivamente. Dessa forma, foram obtidos os *datasets* **D1b**, **D2b**, **D3b** e **D4b**, que representam os mesmos cenários de tráfego sob diferentes condições de latência.

Após a coleta, os dados passaram por uma etapa de pré-processamento para consolidar os relatórios de latência individuais de cada caminho em uma base temporal comum, contendo a latência de todos os caminhos para um mesmo *timestamp*. A ordem dos caminhos define o seu respectivo *pathId*. Para cada instante, o *pathId* associado à menor latência é definido como o rótulo a ser predito. Assim, as classes do problema correspondem ao conjunto de *pathIds* do *dataset*.

#### 4. Avaliação de modelos de AM para escolha do melhor caminho

Para avaliar a viabilidade da aplicação de AM na escolha preditiva do melhor caminho, foram conduzidos experimentos utilizando os *datasets* gerados no ambiente emulado. O objetivo principal foi analisar não apenas quais modelos apresentam as melhores métricas, mas compreender como cada algoritmo reage a diferentes características da rede e sua capacidade de generalização diante de mudanças.

A tarefa de seleção de rotas foi modelada como um problema de classificação supervisionada, onde o objetivo do algoritmo é prever, a cada instante de tempo, o *pathId* da rota que apresenta a menor latência. Para os experimentos, foram selecionados sete algoritmos clássicos de AM: Regressão Logística (RL), K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Árvore de Decisão (AD), Extra Trees (ERT), Random Forest (RF) e Gaussian Naive Bayes (GNB).

O *pipeline* de execução adotou uma divisão do conjunto de dados na proporção de 80% para treinamento e 20% para teste. Para todos os algoritmos, foram utilizados os hiperparâmetros padrão da biblioteca *scikit-learn*. A fim de evitar *overfitting*, foi aplicada validação cruzada estratificada no conjunto de treino, cujo treinamento foi descartado após a obtenção das métricas. O desempenho dos classificadores foi avaliado usando as métricas de acurácia, precisão, *recall*, *f1-score* e a análise de matrizes de confusão.

A avaliação da capacidade preditiva dos classificadores fundamenta-se na análise de dois cenários distintos de estado da rede, conforme apresentado na Tabela 2. Devido à restrição de espaço, apenas as matrizes de confusão dos modelos de melhor e pior desempenho (AD, RF, GNB e SVC, respectivamente) para os *datasets* de *streaming* são apresentados nas Figuras 6 e 7. Esse subconjunto foi selecionado por apresentar resultados similares aos obtidos para outros cenários de tráfego, e também permitir a análise de todos os experimentos realizados nesse trabalho.

##### 4.1. Cenário 1: Treino e teste no mesmo *dataset* (Figura 6)

Primeiramente, os modelos foram avaliados quando treinados e validados com recortes de um mesmo *dataset*. Para o *dataset* D1, cujo caminho de menor latência é predominantemente o Amarelo, todos os modelos alcançaram acurácia superior a 99%. Para outros

Tabela 2. Experimentos realizados em cada cenário de teste

Cenário	Experimento	Treino	Teste
(1) Treino e Teste com mesmo dataset	Datasets com todos os caminhos	D1 Vazia	D1 Vazia
		D2 Poisson	D2 Poisson
		D3 Iperf	D3 Iperf
		D4 Streaming	D4 Streaming
	Datasets sem caminhos dominantes	D3a Iperf	D3a Iperf
		D4a Streaming	D4a Streaming
(2) Treino e Teste com datasets distintos	Enlaces com latências originais	D1 Vazia	D2 Poisson, D3 Iperf, D4 Streaming
	Enlaces com latências modificadas	D1 Vazia	D2b Poisson, D3b Iperf, D4b Streaming

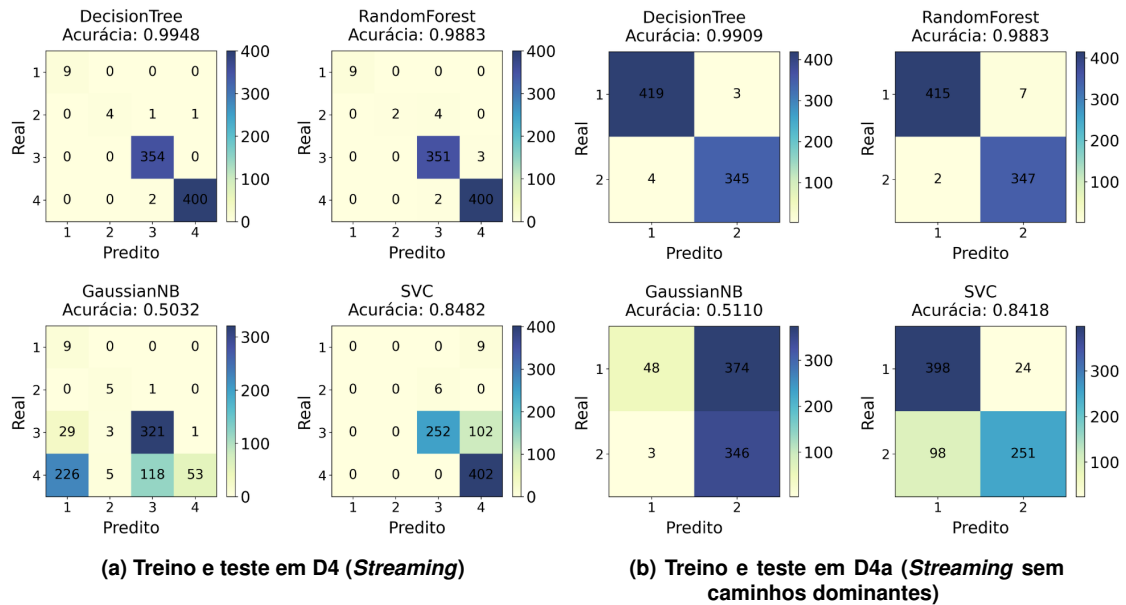


Figura 6. Cenário 1: Matrizes de Confusão para dataset Streaming.

datasets, os modelos de RL e os baseados em árvores de decisão apresentam valores acima de 90%. Nesse contexto, apresentar baixas taxas de *recall* e *f1-score* para as outras classes do problema não caracteriza uma falha relevante, pois a escolha desse caminho implicaria direcionar fluxos para uma rota que, na maior parte do tempo, apresenta latência superior.

Os algoritmos SVC e GNB, por sua vez, apresentam dificuldade de generalização em datasets com maior variabilidade de rótulos. Enquanto o SVC tende a generalizar o conjunto para prever apenas a classe dominante, o GNB parece não ter encontrado o padrão presente nos dados. Dessa forma, esse modelo apresenta a menor acurácia em todas as previsões, chegando a 36% para o dataset D3, e 50% para o D4.

Quando avaliados nos datasets D3a e D4a (sem caminhos dominantes), os modelos de RL e baseados em árvores de decisão apresentam acurácias superiores a 97%, indicando capacidade de capturar o comportamento dinâmico da telemetria e selecionar

o caminho de menor latência na maioria dos instantes. Em contraste, os modelos SVC e GNB mantêm desempenho inferior, com *recall* de 52% e 56% para o caminho Vermelho, menos frequente. Esse comportamento sugere uma tendência à generalização em favor da classe mais ocorrida, falhando em acompanhar de forma consistente as transições entre as duas rotas concorrentes.

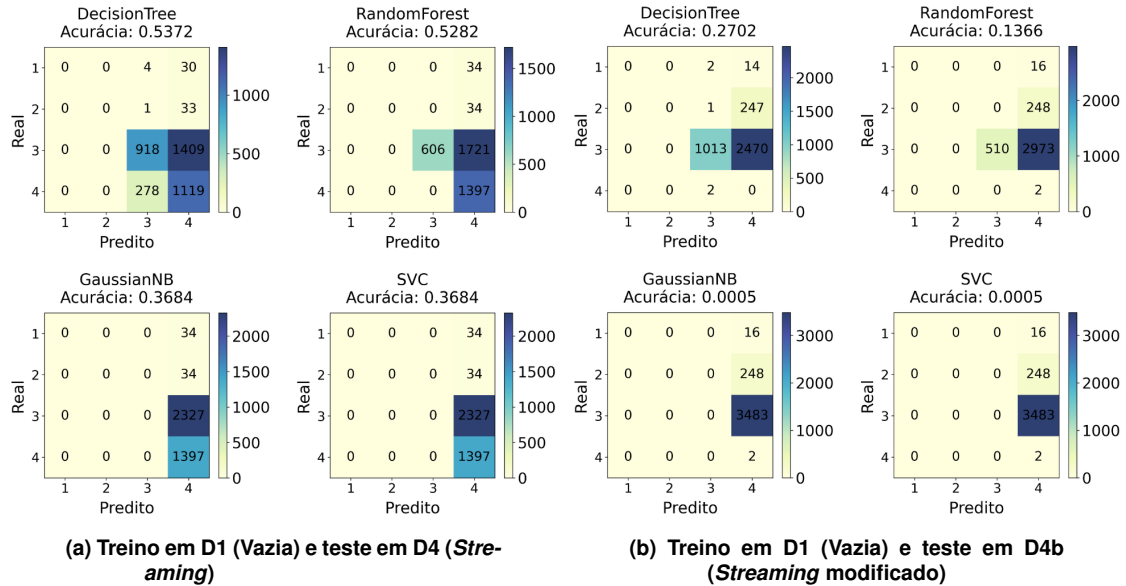


Figura 7. Cenário 2: Matrizes de Confusão para *dataset Streaming*.

#### 4.2. Cenário 2: Treino e teste com *datasets* diferentes (Figura 7)

Quando treinados com o *dataset* D1 (rede vazia) e validados em outros cenários, todos os modelos apresentaram queda consistente de desempenho. No *dataset* D2, as acurácias permaneceram em torno de 83%, tendo a AD o pior desempenho (77%). A maioria dos modelos concentrou as previsões no caminho Amarelo, resultando em precisão de 0% para os caminhos Verde e Vermelho. Em síntese, os classificadores não conseguiram transferir o conhecimento do estado ocioso para cenários com tráfego, evidenciando que os modelos apenas repetem a dominância das classes aprendidas. Esses resultados reforçam a necessidade de retreinamento diante de mudanças no padrão de tráfego.

No último caso de teste, os modelos treinados com o *dataset* D1 foram avaliados nos *datasets* com latências modificadas (D2b, D3b e D4b), resultando em acurácia de 0% na maioria dos casos. Mesmo no *dataset* D4b, onde alguns algoritmos alcançaram entre 10% e 35%, os valores são compatíveis com acertos ocasionais, sem evidência de generalização consistente. Após a modificação das latências, o caminho de menor custo passou a ser o Azul; contudo, os modelos continuaram prevendo majoritariamente o caminho Amarelo, dominante no conjunto de treino, caracterizando *overfitting*.

### 5. Conclusões e Trabalhos Futuros

Este trabalho investigou a viabilidade da aplicação de algoritmos de AM supervisionado para apoiar as decisões de ET em redes cientes de caminho. Inicialmente, a análise de dados públicos evidenciou que *datasets* coletados em redes tradicionais apresentam

limitações intrínsecas que inviabilizam sua aplicação para o treinamento de modelos preditivos orientados a caminhos. Para superar esse obstáculo, um protótipo emulado baseado em SDN foi desenvolvido, capaz de gerar *datasets* de telemetria perfeitamente alinhados, consistentes e específicos para múltiplos caminhos sob diferentes condições de tráfego.

A avaliação experimental demonstrou que algoritmos clássicos, como Árvore de Decisão e Regressão Logística, atingem acurácias superiores a 97% em cenários estáticos ou previamente conhecidos. Contudo, os modelos mostram-se incapazes de realizar generalização frente a modificações estruturais nas latências dos enlaces ou novos padrões de tráfego, situações nas quais o desempenho dos modelos degradou-se severamente, atingindo 0% de acurácia. Tal comportamento evidencia que os classificadores tendem a memorizar a classe predominante do conjunto de treinamento em vez de decodificar a regra implícita da menor latência. Portanto, conclui-se que a viabilidade da aplicação de modelos de AM em ambientes de rede dinâmicos exige a implementação de mecanismos de atualização e retreinamento contínuo para sustentar a eficácia das predições.

Na arquitetura implementada, o controlador Ryu atua principalmente na atualização das regras de encaminhamento, enquanto o processamento de AM é delegado a um agente externo que interage com o controlador por meio de interfaces de aplicações, realizando consultas periódicas assíncronas. Essa separação transfere o custo computacional do treinamento para fora do plano de controle, preservando o desempenho da rede. Assim, o retreinamento dos modelos justifica-se quando seu tempo de convergência é inferior ao ganho de latência obtido ou quando a predição permanece eficaz por tempo suficiente para superar a resposta dos métodos tradicionais de engenharia de tráfego.

Dessa forma, este trabalho estabelece uma base experimental concreta para o avanço da AM na ET em redes cientes de caminho, evidenciando tanto o potencial quanto os limites dos classificadores clássicos. Para mitigar a memorização e aumentar a generalização, trabalhos futuros focarão em modelos como GNN e DRL, aliados a técnicas de aprendizado *online*, janelas deslizantes e detecção de *drift*. Complementarmente, busca-se validar as propostas em topologias não usadas na etapa de treinamento sob tráfego heterogêneo realista. Para enriquecer os modelos de AM, planeja-se expandir as métricas de QoS coletadas via telemetria, como utilização do enlace e tamanho de filas, e aplicar uma análise de importância de atributos. Por fim, a pesquisa avançará rumo à integração das decisões de roteamento com o plano de controle, explorando o roteamento de fonte e o uso de roteamento preditivo.

## Agradecimentos

Este trabalho foi realizado com apoio financeiro parcial da FAPES (PIBICES, 732/2024, 1003/2025 e 2025-1H3FP, 941/2022, 732/2024), FAPESP (PORVIR-5G 20/05182-3) e do CNPq, fellow (Grant 312058/2023-3).

## Referências

- Al-Najjar, A. et al. (2024). Framework for integrating machine learning methods for path-aware source routing. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 829–838.
- Alekseeva, D. et al. (2021). Comparison of machine learning techniques applied to traffic prediction of real wireless network. *IEEE Access*, 9:159495–159514.

- Bernárdez, G. et al. (2021). Is machine learning ready for traffic engineering optimization?
- Comitê Técnico de Monitoramento de Redes – RNP (2024). Especificação do primeiro data challenge. [https://drive.google.com/file/d/1EQm\\_2uubv6H1QxpNis8rYSrRCWaVkJ4E/view](https://drive.google.com/file/d/1EQm_2uubv6H1QxpNis8rYSrRCWaVkJ4E/view).
- Contributors, M. P. (2025). Mininet: An instant virtual network on your laptop (or other pc). <https://mininet.org/>. Acesso em: 18 nov. 2025.
- Farrel, A. (2024). Overview and Principles of Internet Traffic Engineering. RFC 9522.
- Ferreira, G. O. et al. (2023). Forecasting network traffic: A survey and tutorial with open-source comparative evaluation. *IEEE Access*, 11:6018–6044.
- Hartert, R. et al. (2015). A declarative and expressive approach to control forwarding paths in carrier-grade networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM '15*, page 15–28, New York, NY, USA. Association for Computing Machinery.
- International Telecommunication Union (2023). Internet traffic.
- Kiran, M., Campbell, S., and Buraglio, N. (2022). Hecate: Ai-driven wan traffic engineering for science. In *2022 IEEE/ACM International Workshop on Innovating the Network for Data-Intensive Science (INDIS)*, pages 41–49.
- Kubo, R. et al. (2014). Ryu sdn framework-open-source sdn platform software. *NTT Technical Review*, 12.
- Kurose, J. F. and Ross, K. W. (2017). *Computer Networking: A Top-Down Approach*. Pearson, Boston, MA, 7th edition.
- McKeown, N. et al. (2008). Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- Mestres, A. et al. (2017). Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review*, 47(3):2–10.
- Paraíso, D. (2023). Simulação de performance de serviços de comunicação de dados pela plataforma GNS3. Master's thesis, Instituto Federal do Espírito Santo, Vitória, ES.
- Paullada, A. et al. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11).
- Sandvine (2024). Global Internet Phenomena Rreport March 2024. Technical Report v20240213, Sandvine Corporation, 5800 Granite Parkway, Suite 170, Plano, TX 75024, USA.
- Trammell, B. (2022). Current Open Questions in Path-Aware Networking. RFC 9217.
- Yadav, A. (2018). Network path prediction and selection using machine learning. US Patent App. 15/620,247.
- Yu, M. (2019). Network telemetry: towards a top-down approach. *SIGCOMM Comput. Commun. Rev.*, 49(1):11–17.
- Zhou, F. et al. (2025). Traffic engineering in large-scale networks with generalizable graph neural networks.