

# Predição de Qualidade de Serviço em Redes V2X Utilizando Técnicas de Aprendizado de Máquina

Luis Alves Correia Filho, André Luiz Carvalho Ottoni e Marcela Silva Novo

**Resumo**—As redes de comunicação Vehicle-to-Everything (V2X) são essenciais para integrar sistemas de transporte de forma inteligente. Neste contexto, a previsão de parâmetros de qualidade de serviço (QoS) auxilia na tomada de decisões para melhorar a conectividade e garantir a segurança operacional. Este trabalho consiste em analisar técnicas de aprendizado de máquina aplicadas à predição de QoS em redes V2X. Os experimentos foram realizados em um conjunto de dados público considerando a separação dos dados por área e rodadas de medição. Os resultados indicam que o *gradient boosting* apresenta o melhor desempenho entre os algoritmos analisados tanto do ponto de vista de métricas de erro quanto de custo computacional.

**Palavras-Chave**—inteligência artificial, aprendizado de máquina, veículo-para-tudo, qualidade de serviço, transporte inteligente

**Abstract**—Vehicle-to-Everything (V2X) communication networks are essential for integrating smart transportation systems. In this context, predict quality of service (QoS) parameters helps in decision-making to improve connectivity and ensure operational security. This work consists of analyzing machine learning techniques applied to QoS prediction in V2X networks. The experiments were carried out on a public dataset considering the separation of data by area and measurement rounds. The results indicate that gradient boosting presents the best performance among the algorithms analyzed both from the point of view of error metrics and computational cost.

**Keywords**—artificial intelligence, machine learning, vehicle-for-everything, quality of service, smart transportation.

## I. INTRODUÇÃO

Com a modernização do transporte, há a necessidade crescente de comunicação do tipo “veículo-para-tudo” (V2X), que trata da conexão entre diversos tipos de veículos, elementos de infraestrutura de trânsito, redes sem fio e pedestres. Esse tipo de conexão tem despertado grande interesse graças ao potencial de melhorar a segurança no trânsito e possibilitar novos serviços relacionados a sistemas de transporte inteligentes, como direção remota e autônoma [1]. Nesse cenário, é necessário ter um sistema de comunicação de alta confiabilidade, que satisfaça a requisitos rigorosos de qualidade de serviço (QoS) [2], apresentando uma baixa latência e alta taxa de transmissão de dados.

Para garantir a segurança dos veículos e seres humanos no trânsito, a previsão de parâmetros de QoS oferece grandes ganhos para a operação. Se for detectada alguma queda ou alteração dos recursos de comunicação, é possível que a

Luis Alves Correia Filho, Coordenação de Engenharia Elétrica, IFBA, Vitória da Conquista - BA, e-mail: luis.correia@ifba.edu.br; André Luiz Carvalho Ottoni, Departamento de Computação, UFOP, Ouro Preto-MG, e-mail: andre.ottoni@ufop.edu.br; Marcela Silva Novo, Departamento de Engenharia Elétrica, UFBA, Salvador-BA, e-mail: marcela.novo@ufba.br

aplicação se adapte para continuar funcionando de forma adequada, evitando acidentes e prejuízos na sua operação [3].

Em [4] é feita a predição de QoS considerando vários fatores relevantes, como a seleção de recursos utilizados para predição de taxa de transmissão de dados. Também em [5], é mencionado que as características dos ambientes presentes nas medições realizadas devem ser levadas em consideração para diminuir o erro na predição.

O objetivo desse trabalho é trazer uma avaliação de desempenho de algoritmos de aprendizado de máquina na predição de taxa de transmissão de dados em um conjunto de dados recente, e verificar a diferença nos resultados a partir de estratégias de separação de dados conforme áreas e rodadas de medição.

O trabalho está organizado da seguinte forma: a seção II trata de uma breve descrição das técnicas de aprendizado de máquina que serão utilizadas nos experimentos, a seção III descreve a configuração dos experimentos realizados, a seção IV evidencia os resultados obtidos nos experimentos e a seção V traz as considerações finais do trabalho.

## II. TÉCNICAS DE APRENDIZADO DE MÁQUINA

Diversos algoritmos de aprendizado de máquina foram desenvolvidos, sendo que cada um se torna mais adequado para cada aplicação. As árvores de decisão são estruturas muito utilizadas em várias aplicações de machine learning. Estas são compostas de um primeiro nó (raiz) que é ligado a nós intermediários, até os nós terminais (folhas) que irão conter as possíveis saídas da árvore de decisão. Cada trajeto da raiz até as folhas é feita a partir de uma sequência de decisões tomadas [6]. Um exemplo de árvore de decisão é representada na Fig. 1.

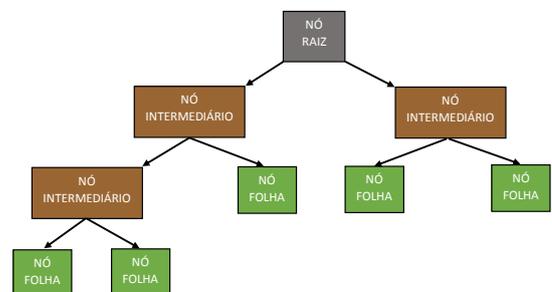


Fig. 1. Ilustração de uma árvore de decisão de cinco folhas e três níveis de profundidade. Fonte: baseado em [6]

É possível combinar múltiplas árvores de decisão para ter modelos mais robustos e que tenham um potencial de

generalização maior do que uma única árvore. Dois exemplos de métodos que combinam múltiplas árvores de decisão são: floresta aleatória (*Random Forest* - RF) e aumento de gradiente (*Gradient Boosting* - GB).

O método de RF utiliza várias árvores de decisão, treinando cada uma com recursos diferentes selecionados aleatoriamente, e para obter a saída, é necessário combinar as saídas individuais de cada modelo para encontrar um único resultado, a partir de uma média aritmética simples ou ponderada [6].

O algoritmo GB cria uma corrente de árvores de decisão de maneira progressiva, sendo este construído em etapas. O início do treinamento é feito em um modelo base, e depois em modelos posteriores, de forma que dados classificados ou estimados de maneira incorreta em um modelo anterior, vão sendo ajustados nos modelos subsequentes [6].

O algoritmo Perceptron Multicamadas (*Multilayer Perceptron* - MLP) é também muito utilizado para resolução de problemas de ML, e se trata de um conjunto de neurônios distribuídos em camadas ocultas, sendo que cada neurônio transforma os valores das camadas anteriores em uma soma linear com pesos, seguida por uma função de ativação [6]. Um MLP de 2 camadas ocultas é representado na Fig. 2.

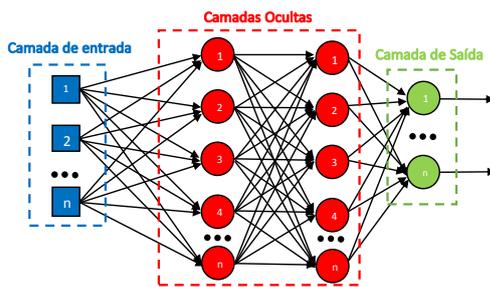


Fig. 2. Ilustração de um MLP de duas camadas ocultas. Fonte: baseado em [6]

### III. CONFIGURAÇÃO DOS EXPERIMENTOS

Os experimentos foram feitos no ambiente do Google Colab em uma máquina com processador Intel(R) Xeon(R) CPU 2.20GHz e 13 GB de memória RAM.

As etapas do experimento estão ilustradas no diagrama de blocos da Fig. 3.

#### A. Dataset Utilizado e Pré-processamento

Para os experimentos, foi utilizado o conjunto de dados Berlin V2X [7], que inclui dados obtidos de veículos ao longo de uma rota de medição seguindo os modos de condução: pelotão (quatro veículos em comboio) e 2X2 (duas duplas de veículos), totalizando 17 voltas na rota pré-estabelecida.

A rota de medição inclui trechos da cidade de Berlin que tem características diferentes, sendo cinco áreas identificadas: residencial, parque, avenida, túnel e autoestrada. Por conta dessas características, esse conjunto de dados oferece a possibilidade de estudo de comportamento de conectividade entre veículos em locais com diferentes regiões de propagação.

Para aplicação nos algoritmos foram aplicados os seguintes filtros:

- Utilizar medições de downlink e que utilizem como parâmetro de QoS a taxa de transmissão de dados
- Remover medições incompletas que não tenham taxa de transmissão de dados
- Remover medições estacionárias (onde a velocidade é igual a 0 km/h) ou que não informem velocidade

O conjunto de dados original apresenta 207434 pontos. Após os filtros, ficaram 32060 pontos para a análise subsequente. O valor da taxa de transmissão de dados também foi dividida por  $10^6$  para ser expressa em Megabits por segundo.

Após filtrar os dados, os recursos (*features*) selecionados para a predição de taxa de transmissão de dados foram:

- camada física (PHY): Relação Sinal Ruído (SNR), Potência recebida do sinal de referência (RSRP), Qualidade recebida do sinal de referência (RSRQ) e Indicador de intensidade do sinal recebido (RSSI)
- Canais físicos compartilhados: blocos de recursos atribuídos, tamanho do bloco de transporte e esquema de modulação e codificação (MCS)
- Controle de recursos de rádio (RRC): identidade da célula, frequência e largura de banda

Após a seleção dos recursos, foi feita uma separação dos dados no formato aleatório, sendo que 80% ficou para treinamento e 20% para teste.

Considerando que o conjunto de dados apresenta medições em diferentes áreas da cidade (avenida - 7066 pontos, autoestrada - 11698 pontos, túnel - 189 pontos, parque - 11698 pontos e residencial - 8580 pontos), os modelos foram treinados, testados e avaliados em cada uma para verificar os resultados em diferentes ambientes de propagação.

Os modelos também foram treinados e testados por rodada de medição. Considerando que o dataset consiste de 17 rodadas em torno de um percurso destacado na Fig. 4, 14 rodadas foram utilizadas para treinamento e 3 para teste.

Para o treinamento do MLP foi feita uma padronização dos recursos na entrada do algoritmo, a partir da remoção da média e dimensionamento os dados para uma variação unitária. Esse procedimento foi feito utilizando a função “*StandardScaler*” da biblioteca sklearn [8] versão 1.5.0.

#### B. Algoritmos e Ajuste de Hiperparâmetros

Os algoritmos foram implementados a partir da biblioteca sklearn [8] versão 1.5.0, disponível para linguagem Python. Foi utilizado o RandomSearchCV dessa mesma biblioteca para otimização dos hiperparâmetros do RF e do GB. Os dados de treinamento definidos foram utilizados para otimização.

Foram escolhidos os hiperparâmetros mencionados em [4] para otimização, sendo estes o número de árvores e a profundidade máxima das árvores, os demais hiperparâmetros foram os definidos na biblioteca padrão do sklearn.

Os hiperparâmetros e o espaço de busca submetidos ao RandomSearchCV foram:

- número de árvores: [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]

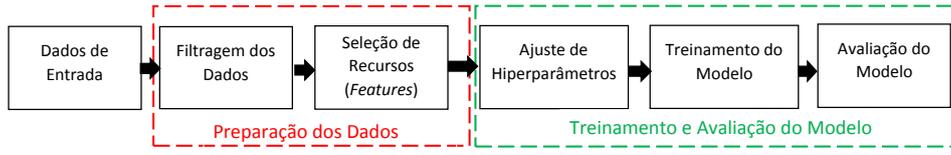


Fig. 3. Diagrama de Blocos ilustrando as etapas dos experimentos.



Fig. 4. Ilustração do percurso feito em uma rodada de medição que constitui o dataset Berlin V2X. Fonte: baseado em [7]

TABELA I  
MELHORES HIPERPARÂMETROS OBTIDOS UTILIZANDO  
RANDOMSEARCHCV

Modelo	RF	GB
Número de Árvores	400	300
Profundidade Máxima	18	2
Tempo de Busca (s)	3362,3	10793,2

- profundidade máxima das árvores: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Os valores de hiperparâmetros selecionados após a otimização podem ser observados na Tab. I.

Para comparação com as outras técnicas baseadas em árvores, foi utilizada uma arquitetura da rede neural utilizada para predição de QoS em redes veiculares presente em [4]. Sendo essa uma arquitetura constituída de quatro camadas ocultas, de 256, 128, 64 e 32 neurônios respectivamente. Foi treinada com o otimizador Adam, uma taxa de aprendizado de 0,001 e tamanho de lote de 16.

### C. Métricas de Análise

Considerando que os algoritmos foram testados nos experimentos para verificar a sua eficácia na predição de QoS, estes foram aplicados para resolução de um problema de regressão. Sendo assim, foram utilizadas métricas adequadas para avaliar o desempenho.

As métricas de erro utilizadas para avaliação foram: o  $R^2$ , o Erro Médio Absoluto (MAE) e a Raiz do Erro Médio Quadrático (RMSE).

O coeficiente de determinação  $R^2$  é uma métrica importante de avaliação pois avalia o nível de ajuste de um modelo de ML aos dados utilizados no treinamento. Essa métrica é calculada

utilizando o valor real ( $y$ ), o valor predito ( $\hat{y}$ ) e a média dos valores reais ( $\bar{y}$ ) conforme (1) [9].

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (y_j - \bar{y})^2} \quad (1)$$

O MAE é comumente utilizado para avaliar modelos de regressão, sendo uma métrica direta da diferença entre os valores reais e os valores preditos [9]. Essa métrica é calculada conforme (2).

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{j=0}^{n-1} |y_j - \hat{y}_j| \quad (2)$$

O RMSE é uma métrica complementar ao MAE para avaliar o desempenho e é mais sensível a valores discrepantes do que o MAE [9]. Essa métrica é calculada conforme (3).

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{j=0}^{n-1} (y_j - \hat{y}_j)^2} \quad (3)$$

Outro ponto importante de ser considerado é o custo computacional de cada modelo, para avaliar isso também foram monitorados o tempo necessário para treinamento do modelo e o tempo para realização da predição de QoS.

## IV. RESULTADOS ANALISADOS

Após a realização dos experimentos, nessa seção está a avaliação dos resultados obtidos nas duas subseções seguintes, sendo avaliado o modelo por área e por rodadas de medição.

### A. Resultados por Área

Inicialmente, foram avaliados os modelos treinados para cada área separadamente. Os resultados obtidos para cada área separadamente podem ser observados na Tab. II.

Em todos os casos, o GB performou melhor frente às métricas avaliadas. Os resultados no túnel foram diferentes em comparação com as outras áreas, isso se deve à pequena quantidade de amostras com relação ao total de pontos do dataset.

O MLP teve uma performance intermediária com relação aos outros algoritmos, porém o seu custo computacional para treinamento é superior aos dos demais modelos, tendo o tempo de treinamento até 30 vezes maior do que o segundo mais rápido, como observado na área do parque.

TABELA II

RESULTADOS OBTIDOS COM OS MODELOS PARA PREDIÇÃO DE TAXA DE TRANSMISSÃO DE DADOS CONSIDERANDO CADA ÁREA)

Métricas	RF	GB	MLP
Residencial			
$R^2$	0,939	0,949	0,945
MAE	5,036	4,981	5,400
RMSE	9,435	8,661	9,006
Tempo de Treinamento (s)	38,6	7,4	629,1
Tempo de Predição (s)	0,261	0,011	0,164
Parque			
$R^2$	0,934	0,936	0,936
MAE	5,689	5,595	5,909
RMSE	10,015	9,819	9,813
Tempo de Treinamento (s)	50,0	10,6	1486,4
Tempo de Predição (s)	0,273	0,014	0,339
Túnel			
$R^2$	0,931	0,914	0,976
MAE	3,484	3,935	2,462
RMSE	6,406	7,142	3,735
Tempo de Treinamento (s)	1,2	0,3	4,3
Tempo de Predição (s)	0,029	0,004	0,001
Autoestrada			
$R^2$	0,923	0,926	0,925
MAE	4,891	4,863	5,096
RMSE	8,158	7,976	8,040
Tempo de Treinamento (s)	17,7	3,6	402,5
Tempo de Predição (s)	0,121	0,008	0,234
Avenida			
$R^2$	0,941	0,940	0,935
MAE	5,785	5,984	6,476
RMSE	10,664	10,760	11,154
Tempo de Treinamento (s)	28,8	5,4	465,7
Tempo de Predição (s)	0,175	0,010	0,138

TABELA III

RESULTADOS OBTIDOS COM OS MODELOS PARA PREDIÇÃO DE TAXA DE TRANSMISSÃO DE DADOS (APLICADA EM TODAS AS ÁREAS)

Métricas	RF	GB	MLP
$R^2$	0,939	0,941	0,932
MAE	5,262	5,195	5,750
RMSE	9,460	9,244	9,961
Tempo de Treinamento (s)	159,3	29,0	3299,3
Tempo de Predição (s)	0,885	0,049	0,450

Os modelos também foram treinados e avaliados para todo o conjunto de dados, considerando todas as áreas. Os resultados obtidos para a predição de taxa de transmissão de dados para esse experimento podem ser vistos na Tab. III.

### B. Resultados por Rodadas de Medição

Avaliando o desempenho considerando as rodadas de medição, foram obtidos os resultados da Tab. IV.

Observando esse caso, é possível observar que as métricas de erro obtidas na separação por área foram ligeiramente superiores com relação às rodadas de medição. Sendo assim, utilizar o conjunto de dados separado por área ao invés de por rodadas de medição mostra ser uma estratégia mais interessante para treinar os modelos.

## V. CONSIDERAÇÕES FINAIS

Após a experimentação, dentre os três algoritmos observados, o GB obteve as melhores métricas em ambas as formas de

TABELA IV

RESULTADOS OBTIDOS COM OS MODELOS CONSIDERANDO AS RODADAS DE MEDIÇÃO

Métricas	RF	GB	MLP
$R^2$	0,935	0,936	0,918
MAE	7,041	7,038	8,552
RMSE	11,861	11,808	13,375
Tempo de Treinamento (s)	154,5	28,1	2603,1
Tempo de Predição (s)	1,099	0,036	0,903

divisão dos dados, tanto do ponto de vista de métricas de erro, como de custo computacional. Uma exceção foi observada com relação exclusivamente à área do túnel, que apresenta uma quantidade de amostras no conjunto de dados bem menor do que as outras áreas.

Dentre as duas condições analisadas para os modelos, concluiu-se que treinar e avaliar os modelos com os dados por área apresentou melhores resultados do que se forem avaliados por rodadas de medição.

Analisando o tempo computacional, foi possível observar que o MLP necessitou de muito mais tempo de treinamento e predição do que os outros métodos para ambas as situações observadas.

Para trabalhos futuros, é possível testar os algoritmos para outras métricas de qualidade de serviço e em outros conjuntos de dados, avaliando outros fatores como latência e instabilidade.

## VI. AGRADECIMENTOS

Os autores agradecem ao Instituto Federal de Educação Ciência e Tecnologia da Bahia (IFBA), à Universidade Federal de Ouro Preto (UFOP) e à Universidade Federal da Bahia (UFBA).

## REFERÊNCIAS

- [1] D. C. Moreira, I. M. Guerreiro, W. Sun, C. C. Cavalcante and D. A. Sousa, "QoS Predictability in V2X Communication with Machine Learning", 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5.
- [2] Z. H. Mir, J. Toutouh, F. Filali and Y. -B. Ko, "Enabling DSRC and C-V2X Integrated Hybrid Vehicular Networks: Architecture and Protocol", in IEEE Access, vol. 8, 2020, pp. 180909-180927.
- [3] Barmounakis, Sokratis, et al. "AI-driven, QoS prediction for V2X communications in beyond 5G systems", Computer Networks, 2022, 217, p. 109341.
- [4] A. Palaios et al., "Machine Learning for QoS Prediction in Vehicular Communication: Challenges and Solution Approaches", in IEEE Access, vol. 11, 2023, pp. 92459-92477.
- [5] E. Eyceyurt, Y. Egi, and J. Zec. "Machine-learning-based uplink throughput prediction from physical layer measurements", Electronics v. 11, n. 8, 2022, p. 1227.
- [6] Z. ZHOU. "Machine Learning", Springer Nature, 2021.
- [7] R. Hermángomez et al., "Berlin V2X: A Machine Learning Dataset from Multiple Vehicles and Radio Access Technologies", 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, 2023, pp. 1-5.
- [8] K. Jolly. "Machine learning with scikit-learn quick start guide: classification, regression, and clustering techniques in Python", Packt Publishing Ltd, 2018.
- [9] M. Steurer; R. Hill and N. Pfeifer. "Metrics for evaluating the performance of machine learning based automated valuation models", Journal of Property Research, v. 38, n. 2, 2021, p. 99-129.