

Avaliação da Latência em Sistemas de Monitoramento Veicular Embarcado com Aprendizado de Máquina

Hilton Machado¹, Matheus Andrade¹, Thaís Medeiros¹,
Marianne Silva^{1,2}, Ivanovitch Silva¹

¹ Universidade Federal do Rio Grande do Norte (UFRN), Natal - Brasil

² Universidade Federal de Alagoas (UFAL), Penedo - Brasil

{hilton.machado.088, matheus.diniz.122, thais.araujo.707}@ufrn.edu.br,
marianne.silva@penedo.ufal.br, ivanovitch.silva@ufrn.br

Abstract. *The advancement of automotive technologies and the widespread adoption of OBD-II (On-Board Diagnostics) devices have enabled real-time vehicle data collection, supporting diagnostics and performance analysis applications. Although the OBD-II protocol is standardized, its implementation across different hardware and vehicles can significantly impact communication latency with the Electronic Control Unit (ECU). This study investigates how such variability affects the efficiency of data acquisition and the performance of embedded machine learning algorithms in a mobile application. Through a case study involving multiple vehicle models and data collection devices, ECU response times and algorithm execution times were analyzed under different operating conditions. The results show that system performance is sensitive to vehicle and interface hardware characteristics, highlighting the need to adjust acquisition parameters to ensure reliable and accurate real-time data processing. The findings contribute to improving embedded solutions for smart mobility and vehicle monitoring.*

Resumo. *O avanço das tecnologias automotivas e a popularização dos dispositivos OBD-II (On-Board Diagnostics) têm viabilizado a coleta de dados veiculares em tempo real, favorecendo aplicações em diagnóstico e análise de desempenho. Apesar da padronização do protocolo OBD-II, a forma como ele é implementado em diferentes hardwares e veículos pode impactar significativamente a latência na comunicação com a Unidade de Controle Eletrônico (ECU). Este trabalho investiga a influência dessas variabilidades sobre a eficiência da coleta de dados e o desempenho de algoritmos de aprendizado de máquina embarcado em um aplicativo móvel. Por meio de um estudo de caso com múltiplos modelos de veículos e dispositivos de coleta, foram analisados os tempos de resposta das ECUs e os tempos de execução dos algoritmos embarcados, considerando diferentes contextos de operação. Os resultados demonstram que o desempenho do sistema é sensível às características do veículo e do hardware de interface, evidenciando a necessidade de ajustes nos parâmetros de aquisição para garantir a confiabilidade e a precisão dos dados processados em tempo real. As análises contribuem para o aprimoramento de soluções embarcadas voltadas à mobilidade inteligente e ao monitoramento veicular.*

1. Introdução

O setor automotivo tem passado por uma transformação significativa com a incorporação de tecnologias digitais, conectividade e processamento local de dados [Pérez-Moure et al. 2023]. Essa transformação viabiliza a coleta, análise e utilização de dados em tempo real para suportar decisões automatizadas e oferecer diagnósticos mais precisos. Nesse contexto, o padrão *On-Board Diagnostics* (OBD-II) tem desempenhado papel central ao permitir o acesso padronizado a parâmetros operacionais dos veículos em tempo real [Jung et al. 2024]. Inicialmente voltado à detecção de falhas e emissões, o OBD-II tornou-se um mecanismo multifuncional, viabilizando aplicações em manutenção preventiva, análise de desempenho e comportamento do condutor [Kumar and Jain 2023a, Malik and Nandal 2023].

Diversos dispositivos comerciais implementam o padrão OBD-II e permitem sua integração com sistemas móveis, possibilitando a construção de soluções de monitoramento veicular [Thajudheen et al. 2023, More et al. 2024]. Contudo, mesmo com a padronização do protocolo, observa-se variabilidade nos tempos de resposta dos sensores, atribuída a fatores como o modelo do veículo, o hardware utilizado e as condições operacionais [Costa et al. 2024, Khan et al. 2023]. Tal latência pode comprometer a temporalidade e a precisão das informações, afetando diretamente a eficácia das soluções baseadas em dados [Kumar and Jain 2023b].

Paralelamente, tem crescido o uso de algoritmos de Aprendizado de Máquina (AM) em aplicações automotivas, com foco em predição de falhas, análise de condução e recomendação de ações [Rana and Khatri 2024, Mandala 2024]. A execução embarcada desses algoritmos, utilizando arquiteturas como TinyML, busca reduzir a dependência de conectividade e oferecer respostas em tempo real [Purnomo et al. 2023]. Entretanto, a latência na aquisição dos dados pode impactar o desempenho dos modelos, sobretudo em cenários sensíveis ao tempo. Ainda são limitados os estudos que analisam, de forma integrada, a variabilidade nos tempos de resposta de dispositivos OBD-II e o impacto dessa latência sobre a execução de algoritmos de aprendizado de máquina embarcado [Purnomo et al. 2023, Waisara et al. 2023]. Compreender como as características dos veículos e dos dispositivos influenciam esses tempos é essencial para garantir precisão e eficiência em aplicações em tempo real. Neste contexto, destaca-se a importância de investigar o intervalo ótimo de requisição de dados — ou seja, a periodicidade ideal para coleta sensorial em diferentes combinações de hardware e veículos [Tak and Choi 2022].

Este trabalho propõe uma abordagem integrada de aquisição e análise de dados veiculares, combinando processamento local em dispositivos móveis com transmissão seletiva para a nuvem. A estratégia visa reduzir a latência na coleta e otimizar a execução de modelos embarcados, considerando o tempo de resposta das ECUs e o custo computacional dos algoritmos. A partir de um estudo de caso com diferentes veículos e dispositivos OBD-II, analisam-se os fatores que influenciam o desempenho temporal do sistema de monitoramento veicular.

O restante do artigo é organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a abordagem proposta. A Seção 4 apresenta o estudo de caso. Na Seção 5, são discutidos os resultados obtidos e, por fim, a Seção 6 apresenta as conclusões e perspectivas para trabalhos futuros.

2. Trabalhos Relacionados

O uso de dispositivos compatíveis com o padrão OBD-II para coleta de dados veiculares tem sido amplamente explorado, principalmente em aplicações de diagnóstico e monitoramento de desempenho. De acordo com [Roque et al. 2024], o OBD-II oferece uma interface padronizada para acessar dados da ECU, sendo amplamente adotado em ferramentas de diagnóstico automotivo. No entanto, a variabilidade nos tempos de resposta entre diferentes dispositivos pode comprometer a qualidade e a eficiência da coleta de dados [Ragab et al. 2024].

Paralelamente, algoritmos de Inteligência Artificial têm sido aplicados à análise desses dados com o objetivo de prever falhas, identificar padrões de condução e otimizar o desempenho veicular [Costa et al. 2024]. Estudos como o de [Manoharan et al. 2024] demonstram que técnicas de aprendizado de máquina, ao integrar dados em tempo real de múltiplos sensores, podem aumentar significativamente a precisão dos diagnósticos. No entanto, grande parte da literatura negligencia o impacto da latência na coleta de dados sobre o desempenho desses algoritmos embarcados [Slimani et al. 2025].

Apesar do crescente uso de aprendizado de máquina em sistemas automotivos, são escassas as investigações que comparam diferentes dispositivos OBD-II sob a perspectiva da latência de aquisição e sua influência na execução de modelos embarcados. A literatura carece de análises que considerem o papel conjunto do hardware e do modelo de veículo no desempenho temporal do sistema. Este trabalho busca contribuir nesse sentido, avaliando as diferenças nos tempos de resposta e nos tempos de execução de algoritmos embarcados em diferentes configurações veiculares e de dispositivos.

3. Abordagem Proposta

Nesta seção, descreve-se a arquitetura modular proposta para aquisição, processamento e transmissão de dados veiculares. A abordagem é composta por quatro módulos principais: (1) Requisição de Dados, (2) Envio de dados, (3) Predição e (4) Transmissão para a Nuvem. Cada módulo é responsável por uma etapa específica do fluxo de monitoramento veicular em tempo real, conforme ilustrado na Figura 1. A arquitetura foi projetada para ser flexível e escalável, permitindo adaptações conforme as necessidades de aplicação, desempenho ou segurança.

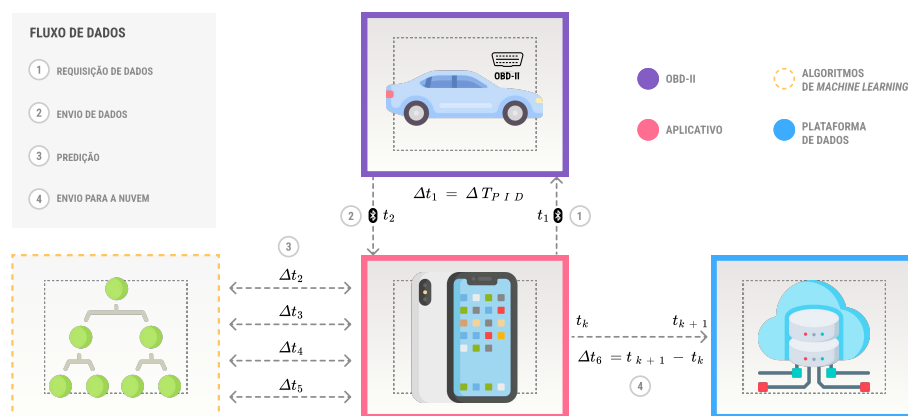


Figura 1. Visão Geral da abordagem proposta.

3.1. Requisição de Dados

A coleta dos dados veiculares é realizada por meio do aplicativo App2Car, desenvolvido em Flutter, com suporte à comunicação com dispositivos ELM327 (microcontrolador ou scanner automotivo) conectados à porta OBD-II dos veículos. Esses dispositivos acessam a rede *Controller Area Network* (CAN) do veículo e traduzem comandos AT e hexadecimais enviados pelo aplicativo para consultas à ECU.

Inicialmente, o App2Car envia uma sequência de comandos AT para configurar o ELM327, seguida dos comandos hexadecimais correspondentes aos *Parameter IDs* (PIDs) – sensores do veículo – requisitados. O dispositivo interpreta as instruções, consulta os sensores, e retorna os dados ao aplicativo, que decodifica e converte as respostas em valores compreensíveis. No total, 27 PIDs foram configurados para coleta, incluindo parâmetros como velocidade, rotação do motor, temperatura, entre outros. Um exemplo da lista de comandos para requisitar a variável de rotações do motor do veículo (RPM) é listada abaixo:

```
// Sequência de configuração e requisição de RPM
ATZ          // Reset do ELM327
ATE0         // Desativa eco
ATL1         // Habilita múltiplas linhas
ATSP0        // Seleção automática de protocolo
010C         // Requisição de RPM
// Resposta esperada: 41 0C 1A F8
// Conversão: (1AF8)_{16} = 6904 => RPM = 6904 / 4 = 1726
```

O tempo total de coleta dos dados pode ser modelado como:

$$T_{\text{PID}} = \sum_{i=1}^N t_{\text{PID}_i} \quad (1)$$

onde T_{PID} representa o tempo total de coleta dos PIDs, N é o número de PIDs requisitados e t_{PID_i} é o tempo necessário para coletar cada PID.

3.2. Envio de Dados

Os dados coletados pelo App2Car são organizados e armazenados localmente em arquivos CSV, gerando dois tipos de registros: (1) *Log Normal* e (2) *Log de Debug*. O *log normal* contém as informações principais requisitadas ao veículo, representando os mesmos dados que serão posteriormente enviados ao servidor. Já o *log de debug* registra informações detalhadas sobre o processo de aquisição, incluindo o sensor requisitado, o tempo de resposta de cada PID e os intervalos entre as requisições. Essa estrutura de registros permite o rastreamento do desempenho do sistema e a análise posterior da eficiência de comunicação e processamento, servindo como base para diagnósticos e validações técnicas.

3.3. Predição

O módulo de predição do App2Car executa quatro algoritmos embarcados de aprendizado de máquina, com o objetivo de identificar padrões no comportamento do condutor e nas condições de operação do veículo. Os modelos são executados sequencialmente após a aquisição dos dados via OBD-II.

Deteção de Outliers (TEDA) – Utiliza o framework TEDA para identificar anomalias com base em medidas de tipicidade e excentricidade [Andrade et al. 2024]. A entrada é a “área do radar”, um sensor virtual calculado a partir da combinação dos sinais de velocidade, rotação do motor (RPM), carga do motor e posição do acelerador.

Classificação do Comportamento do Condutor (MMCloud) – Modelo incremental de agrupamento online que classifica a condução em perfis como “Agressivo”, “Normal” ou “Cauteloso” [Silva et al. 2023, Medeiros et al. 2024]. Utiliza como entradas a área do radar e a carga do motor.

Classificação do Tipo de Combustível (Random Forest) – Identifica se o veículo utiliza como combustível gasolina ou etanol, com foco em aplicações voltadas a emissões veiculares [Amaral et al. 2024]. As variáveis de entrada incluem a porcentagem de etanol, avanço da ignição, velocidade, RPM, posição do acelerador e carga do motor.

Classificação do Tipo de Via (Random Forest) – Determina se o trajeto ocorre em área urbana ou rodoviária, útil para análises de consumo e emissões. As entradas incluem a magnitude do acelerômetro nos eixos X, Y e Z, além de RPM, velocidade, carga e posição do acelerador.

O tempo total de execução dos algoritmos é calculado como:

$$T_{\text{alg}} = \sum_{j=1}^M t_{\text{alg}_j} \quad (2)$$

onde:

- T_{alg} representa o tempo total de execução dos algoritmos embarcados;
- M é o número total de algoritmos executados;
- t_{alg_j} é o tempo de execução do algoritmo j .

O tempo total desde a coleta até a predição é dado por:

$$T_{\text{totalreq}} = T_{\text{PID}} + T_{\text{alg}} \quad (3)$$

onde:

- T_{totalreq} representa o tempo total entre a requisição dos dados dos sensores e a conclusão da execução dos algoritmos de predição;
- T_{PID} é o tempo total para coleta dos dados via OBD-II;
- T_{alg} é o tempo total de execução dos algoritmos embarcados (conforme definido anteriormente).

3.4. Envio de Dados para a Nuvem

O App2Car inclui um módulo dedicado à transmissão periódica dos dados para a nuvem, com o objetivo de garantir persistência, integridade e disponibilidade para análises posteriores. Essa comunicação é realizada via protocolo HTTPS, assegurando a confidencialidade e a integridade das informações transmitidas.

O processo de envio é gerido por três mecanismos complementares:

Orquestração do Envio – Controla a geração e o disparo das requisições HTTP em intervalos regulares.

Monitoramento da Conectividade – Garante que os dados só sejam transmitidos quando houver conexão estável com a internet.

Gerenciamento Seguro da Comunicação – Administra os cabeçalhos, autenticação e tratamento das respostas do servidor, com retentativas automáticas em caso de falhas.

A latência de envio entre dois instantes consecutivos t_k e t_{k+1} é dada por:

$$T_{\text{envio},k} = t_{k+1} - t_k \quad (4)$$

onde:

- $T_{\text{envio},k}$ é o tempo de envio dos dados no ciclo k ;
- t_k é o instante de início da transmissão no ciclo k ;
- t_{k+1} é o instante de término da transmissão no ciclo k .

O tempo total do processo até o instante t_{k+1} é dado por:

$$T_{\text{final},k} = T_{\text{PID},k} + T_{\text{alg},k} + T_{\text{envio},k} \quad (5)$$

onde:

- $T_{\text{final},k}$ representa o tempo total do ciclo k , incluindo aquisição, execução e envio;
- $T_{\text{PID},k}$ é o tempo de coleta dos dados via OBD-II no ciclo k ;
- $T_{\text{alg},k}$ é o tempo de execução dos algoritmos embarcados no ciclo k ;
- $T_{\text{envio},k}$ é o tempo de transmissão dos dados no ciclo k .

Neste artigo, o foco está na análise do tempo total necessário para a obtenção dos dados via OBD-II e para execução embarcada dos algoritmos de predição, conforme definido na Equação 3. Esse tempo representa a latência do sistema até a geração de uma resposta local, sem considerar o envio para a nuvem.

4. Estudo de Caso

Esta seção apresenta um estudo de caso com o objetivo de avaliar a eficiência e a confiabilidade da arquitetura proposta para coleta e predição embarcada de dados veiculares. A investigação foi guiada pelas seguintes questões de pesquisa:

- Q1:** Qual é o intervalo ótimo para a requisição dos dados dos sensores em veículos distintos?
- Q2:** Existe uma diferença significativa no tempo de resposta das ECUs entre os diferentes veículos e dispositivos OBD-II (A e B)?
- Q3:** Os algoritmos embarcados apresentam variação significativa no tempo de execução entre diferentes veículos e dispositivos?
- Q4:** Como se comportam as distribuições dos tempos de resposta dos sensores para os mesmos veículos, utilizando diferentes dispositivos OBD-II?

4.1. Instrumentação

Foram utilizados dois dispositivos OBD-II da linha ELM327: a versão 1.5 (denominada A) e a versão 2.1 (denominada B). A comunicação com os dispositivos foi realizada via Bluetooth, utilizando um smartphone Samsung Galaxy S20 FE com o aplicativo App2Car instalado. A seleção dos veículos foi feita com base na disponibilidade e representatividade de modelos comuns no mercado nacional, abrangendo diferentes marcas, anos e configurações. A Tabela 1 apresenta os veículos utilizados, seus protocolos de comunicação e a quantidade de unidades testadas por modelo.

Tabela 1. Características dos veículos utilizados na coleta de dados.

Veículo	Marca	Ano	Protocolo CAN	Unidades de ECU
Polo	Volkswagen	2019	ISO 15765-4	2
T-Cross	Volkswagen	2023	ISO 15765-4	1
HB20	Hyundai	2023	ISO 15765-4	1
Fit	Honda	2012	ISO 15765-4	1

4.2. Preparação

A primeira etapa consistiu em validar a compatibilidade dos dispositivos OBD-II A e B com os veículos da amostra. Foram realizados testes preliminares para assegurar a comunicação com a ECU e a resposta correta aos comandos de leitura dos PIDs. Após essa validação, foi definido um trajeto de aproximadamente 6 km, com trechos urbanos e rodoviários na cidade de Natal/RN. Todos os veículos percorreram a mesma rota sob condições operacionais semelhantes (horário, temperatura e tráfego), assegurando a comparabilidade dos dados.

4.3. Operação

Durante a coleta, foram registrados 27 PIDs, dados de localização (GPS), aceleração nos três eixos e informações sobre o dispositivo OBD-II conectado. Cada veículo realizou o percurso com ambos os dispositivos, permitindo comparação direta entre os tempos de resposta. Os algoritmos embarcados foram executados em tempo real, e seus tempos de execução foram registrados para cada contexto de uso. A análise estatística incluiu os testes de Kolmogorov-Smirnov (KS) e Shapiro-Wilk para verificação da normalidade, além do teste U de Mann-Whitney para comparação entre grupos independentes [Wohlin et al. 2024].

Todos os dados e scripts utilizados no experimento estão disponíveis no repositório GitHub: <https://github.com/conect2ai/SBRC2025-App2Car>, permitindo total reprodutibilidade dos resultados.

5. Resultados e Discussão

Esta seção apresenta os resultados obtidos a partir do estudo de caso descrito na Seção 4, com foco na análise da latência de comunicação com a ECU e do desempenho dos algoritmos embarcados. Os dados coletados foram analisados com o intuito de responder às quatro questões de pesquisa propostas, abordando aspectos como o intervalo ótimo de

requisição dos sensores, a variação nos tempos de resposta dos dispositivos OBD-II, a eficiência computacional dos modelos de aprendizado de máquina e a estabilidade das distribuições de tempo.

Inicialmente, analisaram-se os tempos de resposta dos sensores veiculares, medidos separadamente para os dispositivos OBD-II A (versão 1.5) e B (versão 2.1), conforme apresentados na Tabela 2. Os resultados indicam que o dispositivo A apresentou maior consistência nas medições, com variações menores entre os veículos testados. A mediana variou entre 137 ms (HB20) e 150 ms (T-Cross). Em contraste, o dispositivo B apresentou maior dispersão, com medianas variando de 136 ms (Fit) até 179 ms (Polo Branco), sugerindo instabilidade temporal em certos contextos de operação.

Tabela 2. Medianas para os Dispositivos OBD's A e B.

Veículo	Mediana - OBD-II A (ms)	Mediana - OBD-II B (ms)
Polo Prata	149.00	178.00
Polo Branco	149.00	179.00
Fit	149.00	136.00
T-Cross	150.00	169.00
HB20	137.00	179.00

A presença de valores atípicos foi evidenciada por medidas de assimetria (Skewness) e curtose (Kurtosis), cujos resultados indicaram distribuições fortemente distorcidas em determinados cenários. Destaca-se o caso do T-Cross com o dispositivo A, que apresentou uma Skewness de 21,15 e uma Kurtosis de 815,43 — valores extremamente elevados, sugerindo forte concentração dos dados em torno da mediana com ocorrência ocasional de tempos de resposta muito elevados (Tabela 3). Esse padrão é característico de sistemas sujeitos a instabilidades momentâneas na comunicação com a ECU. Para mitigar esse efeito, a mediana foi adotada como medida central, em substituição à média. Adicionalmente, o teste de Kolmogorov-Smirnov indicou, com p-valores próximos de zero em todos os casos, a rejeição da hipótese de normalidade, corroborando a escolha de métodos não paramétricos nas comparações entre grupos.

Tabela 3. Métricas para o Dispositivo OBD-II A.

Grupo	Skewness	Kurtosis	KS test p-value
Polo Prata	3.61	142.78	0
Polo Branco	-0.68	14.23	2.56×10^{-286}
Fit	-0.53	19.19	0
T-Cross	21.15	815.43	0
HB20	0.08	-1.10	5.29×10^{-234}

De forma complementar à análise das medidas de tendência central, realizou-se a comparação estatística entre os tempos de resposta dos dispositivos OBD-II A e B. Para isso, foi adotado o teste U de Mann-Whitney, apropriado para amostras independentes sem suposição de normalidade, conforme evidenciado previamente pelos testes de Kolmogorov-Smirnov. Os resultados, sumarizados na Tabela 4, indicam que as diferenças

entre os dois dispositivos são estatisticamente significativas para todos os veículos analisados ($p\text{-value} < 0,05$), com destaque para os casos do Fit e do T-Cross, cujos $p\text{-valores}$ extremamente baixos reforçam a magnitude da divergência.

Tabela 4. Teste U de Mann-Whitney entre Dispositivos A e B.

Grupo	Mann-Whitney U p-value
Polo Prata	0
Polo Branco	0
Fit	0
T-Cross	1.83×10^{-123}
HB20	1.83×10^{-5}

Essa discrepância sugere que a escolha do dispositivo OBD-II pode impactar na confiabilidade da coleta em aplicações sensíveis à latência. A diferença observada entre os tempos de resposta pode decorrer de fatores como o *firmware* embarcado no dispositivo, a versão do protocolo suportado ou a capacidade de paralelismo na leitura dos PIDs.

Para aprofundar a análise, foi conduzida uma comparação entre dois veículos de mesmo modelo (Volkswagen Polo), identificados como Polo Branco e Polo Prata. O objetivo foi investigar a influência de variações intra-modelo no comportamento temporal dos sensores. Para o dispositivo OBD-II A, detectou-se uma diferença estatisticamente significativa entre os veículos ($p\text{-value} = 0,02$), enquanto para o dispositivo B não houve evidência de diferença ($p\text{-value} = 0,94$). Esses resultados revelam que, mesmo dentro de um mesmo modelo, pequenas variações de fabricação ou atualizações no sistema eletrônico podem influenciar a latência da ECU — efeito mais sensível quando o dispositivo possui maior fidelidade na coleta (caso do dispositivo A).

Essas análises reforçam a importância de uma caracterização sistemática dos dispositivos e veículos utilizados, sobretudo em aplicações que exigem monitoramento em tempo real e aprendizado de máquina embarcado. A seleção criteriosa do hardware de interface não deve se restringir à compatibilidade, mas também considerar métricas empíricas de desempenho temporal e estabilidade da comunicação.

A análise dos tempos de execução dos modelos de aprendizado de máquina embarcados permitiu observar diferenças relevantes entre os algoritmos utilizados e também entre os veículos testados. As Figuras 2(a–d) apresentam as distribuições empíricas dos tempos de predição para cada um dos quatro modelos implementados no aplicativo App2Car. Entre eles, destaca-se o Modelo de Detecção de Tipo de Combustível, cuja mediana foi de $201 \mu s$, demonstrando elevada eficiência computacional. Já o Modelo de Comportamento do Motorista apresentou maior complexidade, refletida em uma mediana de $299 \mu s$ e em uma distribuição assimétrica, com caudas mais extensas, indicando maior variabilidade nas execuções. O Modelo de Identificação de Vias ocupou posição intermediária, com uma mediana de $168 \mu s$, apresentando distribuição mais concentrada, o que sugere maior estabilidade nos tempos de execução. O Modelo de Detecção de *Outliers*, baseado no framework TEDA, apresentou os menores tempos de execução, com média inferior a $2 \mu s$, destacando-se pela leveza computacional.

A Tabela 5 consolida os tempos médios de execução dos modelos com suas res-

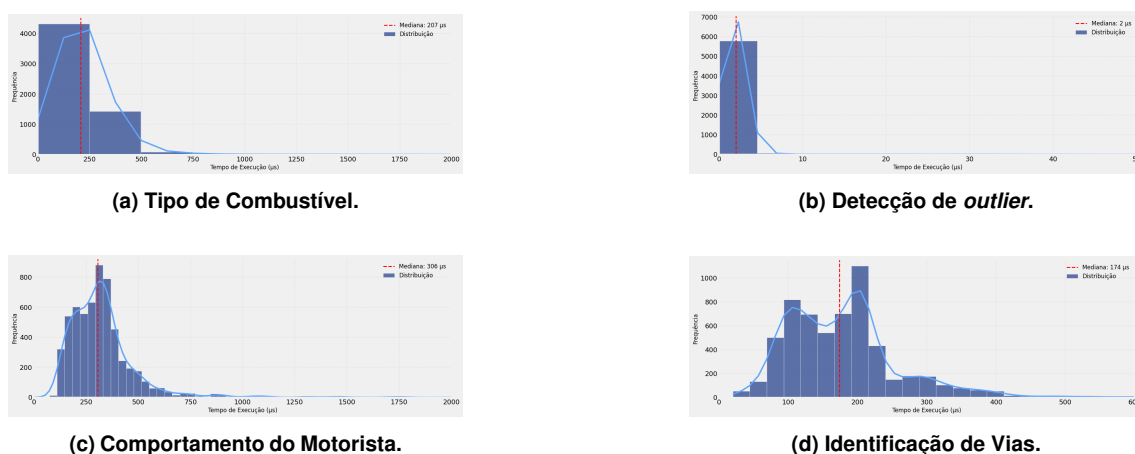


Figura 2. Distribuições dos tempos de execução para os algoritmos de ML.

pectivas incertezas, permitindo uma comparação quantitativa. O tempo total médio para a execução sequencial de todos os modelos embarcados foi de $671.37 \pm 1.95 \mu s$. Esses valores evidenciam que, mesmo em dispositivos móveis com recursos computacionais limitados, é possível realizar inferências em tempo quase real, desde que os modelos sejam otimizados e projetados com foco em baixo custo computacional.

Tabela 5. Tempos médios de execução e suas incertezas para cada modelo.

Modelo	Tempo Médio (μs)
Modelo de Combustível	201.41 ± 1.04
Modelo de Detecção de <i>outlier</i>	1.67 ± 0.01
Modelo de Comportamento do Motorista	299.45 ± 1.40
Modelos de Identificação de Vias	168.84 ± 0.87
Total	671.37 ± 1.95

A Figura 3 apresenta a comparação das distribuições dos tempos de resposta entre os dispositivos OBD-II A e B em diferentes veículos, permitindo uma análise da consistência e estabilidade na comunicação com a ECU. De maneira geral, observa-se que o dispositivo A apresenta distribuições mais concentradas, com menor amplitude e menor dispersão, sugerindo maior previsibilidade nos tempos de resposta. Em contrapartida, o dispositivo B apresenta densidades com caudas mais longas, maior variabilidade e assimetrias acentuadas, indicando maior instabilidade na comunicação.

Destacam-se, em particular, os casos dos veículos T-Cross e Polo Prata, nos quais as diferenças entre os dispositivos são mais pronunciadas. Nestes modelos, o dispositivo B apresenta não apenas tempos medianos mais altos, mas também maior alargamento da curva de densidade, o que evidencia maior latência e inconsistência nas respostas. Por outro lado, chama atenção o fato de que os dois modelos de Polo — Branco e Prata — exibem comportamentos praticamente idênticos quando se comparam os dispositivos A e B, com distribuições muito semelhantes para cada tipo de dispositivo. Deste modo, reforça a hipótese de que o desempenho da comunicação entre o OBD-II e a ECU depende não apenas do modelo do dispositivo, mas também de características específicas de cada

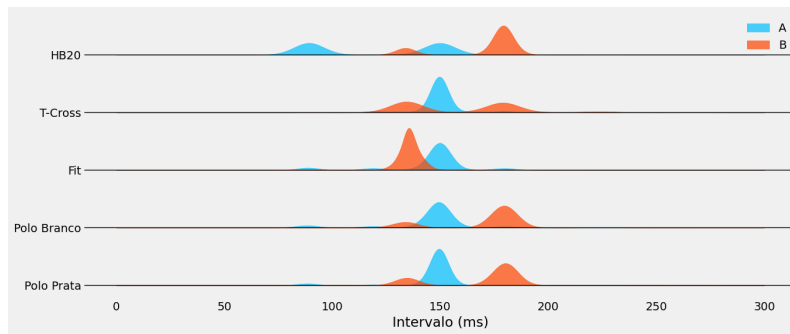


Figura 3. Comparativo de intervalos de requisição para todos os veículos.

veículo. A semelhança nas distribuições dos Polos sugere uma possível homogeneidade na arquitetura eletrônica entre as versões testadas, que pode mitigar os efeitos de variação entre os dispositivos. Assim, os dados evidenciam que a latência na comunicação OBD-II não é apenas uma função do dispositivo utilizado, mas resulta de uma interação entre o hardware de coleta e o sistema embarcado de cada modelo veicular. A análise detalhada dessas distribuições contribui para a compreensão dessa variabilidade e reforça a necessidade de testes multi-ambiente para validação de soluções baseadas em OBD-II.

A Tabela 6 apresenta os tempos totais de requisição de dados T_{total} medidos para cada veículo em dois cenários distintos, representados pelos dispositivos OBD-II A e B. Cada valor é acompanhado do respectivo desvio padrão, permitindo a análise da variabilidade na latência de aquisição.

Tabela 6. Tempo total de Requisições.

Veículo	A (ms)	B (ms)
Polo Prata	3948.83 ± 7.21	4569.39 ± 14.42
Polo Branco	3920.10 ± 6.72	4625.27 ± 13.55
Fit	3935.82 ± 6.01	3691.00 ± 4.34
T-Cross	4057.68 ± 7.63	4355.85 ± 29.33
HB20	3247.02 ± 44.11	4616.78 ± 1.27

Observa-se que o dispositivo A apresentou, de modo geral, menor tempo total de requisição em quatro dos cinco veículos testados. O menor tempo foi registrado no HB20 com 3247.02 ± 44.11 ms, contrastando com o maior tempo observado no T-Cross, que alcançou 4057.68 ± 7.63 ms. Já no cenário com o dispositivo B, os tempos totais foram, em geral, mais elevados, destacando-se o Polo Branco com 4625.27 ± 13.55 ms como o caso de maior latência média. Notavelmente, o veículo Fit apresentou comportamento atípico, com tempo total inferior ao ser utilizado com o dispositivo B (3691.00 ± 4.34 ms) em comparação ao cenário com o dispositivo A (3935.82 ± 6.01 ms). Essa inversão pontual reforça a hipótese de que fatores específicos do hardware veicular ou particularidades da ECU podem interferir na eficiência da comunicação com determinados dispositivos OBD-II.

Tal variação evidencia a importância de estratégias de parametrização dinâmica para os ciclos de aquisição, com vistas à otimização do desempenho do sistema de mo-

nitoramento embarcado. Esses resultados reforçam a hipótese de que o desempenho da comunicação OBD-II não depende apenas do dispositivo em si, mas também de aspectos específicos do veículo, como a implementação da ECU, o protocolo OBD-II utilizado e eventuais limitações físicas ou elétricas do barramento CAN. Dessa forma, evidencia-se a importância de adotar parâmetros de aquisição adaptativos, calibrados conforme o par veículo-dispositivo, a fim de assegurar maior eficiência e estabilidade na coleta de dados. Essa adaptabilidade pode ser especialmente relevante para aplicações embarcadas que operam com restrições de tempo real ou que exigem previsibilidade nos intervalos de aquisição.

Esses achados reforçam a importância de adotar uma abordagem adaptativa para a coleta e o processamento de dados veiculares. Em relação à Q1, verificou-se que o intervalo ideal para a requisição dos sensores varia significativamente conforme o par veículo-dispositivo, o que evidencia a necessidade de estratégias de aquisição dinâmicas e calibradas. Quanto à Q2, os dados revelaram diferenças estatísticas relevantes entre os tempos de resposta das ECUs para distintos veículos e dispositivos, com destaque para a maior variabilidade observada no dispositivo OBD-II B. A análise da Q3 mostrou que os tempos de execução dos algoritmos embarcados de ML são influenciados pela natureza dos modelos e pelas características dos dados, sendo os algoritmos mais complexos — como o de comportamento do motorista — os mais sensíveis a essas variações. Por fim, no tocante à Q4, observou-se que as distribuições dos tempos de resposta são mais concentradas e estáveis no dispositivo OBD-II A, ao passo que o dispositivo B apresenta maior dispersão, o que pode indicar instabilidades na comunicação com a ECU. Esses resultados indicam que o desempenho da comunicação OBD-II não depende apenas do dispositivo utilizado, mas também de aspectos específicos do veículo, como a implementação da ECU, o protocolo de comunicação e as limitações do barramento CAN. Assim, conclui-se que soluções embarcadas eficazes exigem mecanismos de coleta e inferência que se ajustem dinamicamente ao contexto operacional, especialmente em aplicações com restrições de latência, previsibilidade e eficiência energética.

6. Conclusão

Este estudo avaliou o desempenho de um sistema embarcado para coleta e processamento de dados veiculares, utilizando o aplicativo App2Car em conjunto com dispositivos OBD-II. A investigação concentrou-se nos tempos de resposta das ECUs e na eficiência computacional de modelos de aprendizado de máquina executados localmente, considerando diferentes veículos e duas versões de dispositivos (A e B).

Os resultados mostraram que os tempos de resposta variam significativamente conforme o par veículo-dispositivo, sendo o dispositivo OBD-II A mais estável e o B mais suscetível a variações e *outliers*. A análise estatística confirmou essas diferenças e evidenciou a necessidade de calibrar os parâmetros de aquisição com base no contexto. Em relação aos algoritmos embarcados, modelos mais complexos apresentaram maior tempo de execução, reforçando a importância de considerar a carga computacional nas aplicações com restrições de tempo real. As perguntas de pesquisa foram respondidas com base nos dados obtidos: (Q1) o intervalo ideal de requisição depende do veículo e do dispositivo; (Q2) há diferenças significativas entre os tempos de resposta das ECUs; (Q3) os algoritmos de ML respondem de forma distinta a diferentes contextos veiculares;

e (Q4) há variações relevantes nas distribuições dos tempos de resposta entre os dispositivos, impactando a previsibilidade da coleta.

Como direções futuras, destaca-se a ampliação da base de veículos testados, abrangendo diferentes modelos e configurações, com o objetivo de verificar a robustez do sistema frente à variabilidade de assincronias e protocolos de comunicação. Propõe-se também a inclusão de uma análise detalhada do tempo de envio de dados (T_{envio}), por meio da quantificação das latências de transmissão entre dispositivo móvel e servidor. Essas ações viabilizarão uma validação das hipóteses propostas.

Referências

- Amaral, M., Medeiros, M., Andrade, M., Flores, T., Silva, M., and Silva, I. (2024). Tinyml implementation and optimization for fuel type classification on obd-ii edge device. In *2024 Symposium on Internet of Things (SIoT)*, volume 1, pages 1–5. IEEE.
- Andrade, P., Silva, M., Medeiros, M., Costa, D. G., and Silva, I. (2024). Teda-rls: A tinyml incremental learning approach for outlier detection and correction. *IEEE Sensors Journal*.
- Costa, H., Silva, M., Sánchez-Gendríz, I., Viegas, C. M. D., and Silva, I. (2024). An evolving multivariate time series compression algorithm for iot applications. *Sensors*, 24(22).
- Jung, J., Han, S., Park, M., and je Cho, S. (2024). Automotive digital forensics through data and log analysis of vehicle diagnosis android apps. *Forensic Science International: Digital Investigation*, 49:301752.
- Khan, M. A. A., Ali, M. H., Haque, F., and Habib, M. T. (2023). A machine learning approach for driver identification. *Indonesian Journal of Electrical Engineering and Computer Science*, 30(1):276–288.
- Kumar, R. and Jain, A. (2023a). Driving behavior analysis and classification by vehicle obd data using machine learning. *The Journal of Supercomputing*, 79(16):18800–18819.
- Kumar, R. and Jain, A. (2023b). Driving behavior analysis and classification by vehicle obd data using machine learning. *Journal of Supercomputing*, 79(3):18800–18819.
- Malik, M. and Nandal, R. (2023). A framework on driving behavior and pattern using on-board diagnostics (obd-ii) tool. *Materials Today: Proceedings*, 80:3762–3768. SI:5 NANO 2021.
- Mandala, V. (2024). Predictive failure analytics in critical automotive applications: Enhancing reliability and safety through advanced ai techniques. *Journal of Artificial Intelligence and Big Data*, 4(1):48–60.
- Manoharan, M., Muthukrishnan, K., Balan, G., Arumugam, S., Muthusamy, S., Ramachandran, M., Balodi, A., Chinnaiyan, V. K., Sekaran, S., and Gnanakkan, C. A. R. C. (2024). A novel method for illegal driver detection and legal driver identification using multitask learning based lstm models for real time applications. *Wireless Personal Communications*, 136(3):1923–1944.

- Medeiros, M., Flores, T., Silva, M., and Silva, I. (2024). A multi-layered methodology for driver behavior analysis using tinyml and edge computing. In *2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 1–8. IEEE.
- More, A., Khane, S., Jadhav, D., Sahoo, H., and Mali, Y. K. (2024). Auto-shield: Iot based obd application for car health monitoring. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–10.
- Purnomo, C. P., Munadi, R., Istikmal, Widodo, A., Kuntadi, S., and Putra, R. H. (2023). Digitalization of public vehicles using on board diagnostic-ii (obd-ii). In *2023 International Conference On Cyber Management And Engineering (CyMaEn)*, pages 50–54.
- Pérez-Moure, H., Lampón, J. F., Velando-Rodriguez, M.-E., and Rodríguez-Comesaña, L. (2023). Revolutionizing the road: How sustainable, autonomous, and connected vehicles are changing digital mobility business models. *European Research on Management and Business Economics*, 29(3).
- Ragab, H., Givigi, S., and Noureldin, A. (2024). Automotive speed estimation: Sensor types and error characteristics from obd-ii to adas. *arXiv preprint arXiv:2501.00242*.
- Rana, K. and Khatri, N. (2024). Automotive intelligence: Unleashing the potential of ai beyond advance driver assisting system, a comprehensive review. *Computers and Electrical Engineering*, 117:109237.
- Roque, A. D. S., Alves, L. M. D. S., and de Freitas, E. P. (2024). Can-modes: In-vehicle datasets generation and analysis in different driving situations. In *2024 Workshop on Communication Networks and Power Systems (WCNPS)*, pages 1–7. IEEE.
- Silva, M., Medeiros, T., Azevedo, M., Medeiros, M., Themoteo, M., Gois, T., Silva, I., and Costa, D. G. (2023). An adaptive tinyml unsupervised online learning algorithm for driver behavior analysis. In *2023 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*, pages 199–204. IEEE.
- Slimani, C., Morge-Rollet, L., Lemarchand, L., Espes, D., Le Roy, F., and Boukhobza, J. (2025). A study on characterizing energy, latency and security for intrusion detection systems on heterogeneous embedded platforms. *Future Generation Computer Systems*, 162:107473.
- Tak, S. and Choi, S. (2022). Safety monitoring system of cavs considering the trade-off between sampling interval and data reliability. *Sensors*, 22(10).
- Thajudheen, S., G, S., and Jesudoss, A. G. (2023). Vehicular data retrieval system. In *2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*, volume 1, pages 1–5.
- Waisara, S., Charoenlarnnoppa, S., Srisurangkul, C., and Nishio, T. (2023). Vehicle telematics system design for real-time applications using mobile networks. In *2023 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pages 1–6.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2024). *Experimentation in software engineering*. Springer Science & Business Media, 2^a edition.