UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

MMCloud: um algoritmo de *clustering* não supervisionado online para análise do comportamento do motorista

Morsinaldo de Azevedo Medeiros

Orientador: Prof. Dr. Ivanovitch Medeiros Dantas da Silva

Co-orientador: Profa. Dra. Marianne Batista Diniz da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem PPgEEC: M691
Natal, RN, Julho de 2025

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Medeiros, Morsinaldo de Azevedo.
MMCloud: um algoritmo de clustering não supervisionado online para análise do comportamento do motorista / Morsinaldo de Azevedo Medeiros. - 2025.
92 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, 2025.
Orientação: Prof. Dr. Ivanovitch Medeiros Dantas da Silva.
Coorientação: Profa. Dra. Marianne Batista Diniz da Silva.

1. Veículos inteligentes - Dissertação. 2. Soft-Sensors - Dissertação. 3. Agrupamento online incremental - Dissertação. 4. TinyML - Dissertação. 5. Classificação do comportamento do motorista - Dissertação. I. Silva, Ivanovitch Medeiros Dantas da. II. Silva, Marianne Batista Diniz da. III. Título.

RN/UF/BCZM CDU 629

Elaborado por Jackeline dos Santos Pinheiro da Silva Maia Cavalcanti - CRB-15/317

Dedico este trabalho à minha família, em especial aos meus pais e meus avós, pela paciência e pelo apoio durante a realização deste trabalho.

Agradecimentos

Agradeço a Deus pelo dom da vida, por ser a luz que guia meus passos e ilumina meu caminho, mesmo nas horas mais incertas. Sou grato pelas pessoas maravilhosas que Ele me permitiu conhecer e por cada oportunidade vivenciada, que moldou quem sou hoje. Mais do que isso, sou grato pela inteligência que me concedeu e pela força de vontade que me permite perseguir meus objetivos, superar desafios e ir sempre além.

Aos meus pais, Morvanildo dos Santos Medeiros e Rosimar Dantas de Azevedo, minha sincera gratidão. Ao longo da minha trajetória, vocês foram uma base constante de apoio. O carinho, o empenho e a confiança que sempre demonstraram foram fundamentais para meu desenvolvimento pessoal e acadêmico. Agradeço por cada gesto, conselho e oportunidade que contribuíram para minha formação durante a graduação e o mestrado. Obrigado por estarem ao meu lado e por apoiarem meus objetivos.

À minha namorada, companheira e melhor amiga, Maria Luiza Melo Costa de Medeiros, meu mais sincero obrigado. Seu amor, carinho e apoio incondicional foram o meu porto seguro. Você esteve ao meu lado em cada desafio, me encorajando a seguir em frente e se fazendo presente nos momentos mais importantes, celebrando minhas vitórias e me consolando nas minhas dificuldades. Sua presença transformou cada passo dessa jornada em algo mais leve e significativo.

Ao professor Ivanovitch Medeiros Dantas da Silva, deixo meu sincero agradecimento e admiração pelos ensinamentos transmitidos, pelas oportunidades concedidas, pela atenção e pela constante disponibilidade. Suas orientações sempre claras e valiosas, aliadas à postura atenciosa e ao apoio contínuo ao longo de todo o processo, contribuíram de forma significativa para o meu desenvolvimento acadêmico e pessoal. Mais do que um orientador, sua confiança no meu trabalho e seu compromisso com a formação de seus orientandos fizeram diferença em toda a minha trajetória.

À minha co-orientadora, Marianne Diniz, muito obrigado por sua fundamental contribuição na construção deste trabalho. Sua paciência, atenção e disponibilidade foram essenciais. Sou grato por ter me dado a oportunidade de desenvolver um trabalho tão incrível, baseado em sua tese de doutorado. Sua expertise e generosidade foram inspiradoras.

Agradeço aos professores Luiz Affonso Henderson Guedes de Oliveria e Carlos Manuel Dias Viegas por toda a ajuda e os conhecimentos preciosos que generosamente compartilharam.

Aos meus amigos do grupo de pesquisa Conect2ai, em especial à Thaís Medeiros, por

toda a amizade e parceria constante ao longo do mestrado e parte da graduação. Ao Matheus Diniz, pela amizade, pelas risadas, pelas valiosas dicas de treino e alimentação que me ajudaram a manter o equilíbrio, e pelos muitos cafés que tornaram os momentos de estudo mais leves e descontraídos. Ao Miguel Eurípedes, também pela amizade e pelos conselhos valiosos que contribuíram significativamente para a minha rotina. Ao Fellipe Milomem por todas as conversas sobre video games e dicas de treino. E ao Breno Santos, pelas contribuições fundamentais sobre escrita científica, que aprimoraram este trabalho, e pelos cafés que nos acompanharam ao longo da pesquisa.

Aos amigos que conheci durante o projeto do Conselho Nacional de Justiça (CNJ), em especial Gisliany Alves, Alexandre Henrique e Germano Yoneda. Sou muito grato pela rica troca de experiências que tive com vocês.

Por fim, minha gratidão se estende a todos que, direta ou indiretamente, contribuíram para a minha formação e para a conclusão desta importante etapa.

"Um sucesso que se satisfaz é um sucesso que paralisa.
Excelência não é um lugar aonde você chega.
Excelência é um horizonte."

— Mario Sérgio Cortella,
Qual é a sua obra?

Resumo

Sistemas inteligentes evolutivos e adaptativos são fundamentais para lidar com fluxos dinâmicos de dados do mundo real, especialmente em ambientes com restrições de recursos, como veículos habilitados para Internet das Coisas (IoT, do inglês *Internet of Things*). Este estudo propõe uma abordagem de *clustering* evolutivo online para diagnóstico em tempo real do comportamento do motorista, aproveitando *Tiny Machine Learning (TinyML)* e *soft sensors*. Este estudo propõe o algoritmo *MMCloud*, um modelo de *clustering* incremental desenvolvido para lidar com fluxos contínuos de dados. Ele dispensa a necessidade de re-treinamento e é capaz de se adaptar à mudança de conceito (*concept drift*) e às condições variáveis de direção. A metodologia combina diagnóstico a bordo (*OBD-II*) com um *framework* de *edge computing* para classificar o comportamento de condução do motorista em três categorias: cautelosa, normal e agressiva. Para validar a abordagem, dois estudos de caso foram conduzidos em ambientes urbanos com diversas condições de tráfego. Um deles utilizou um dispositivo Freematics One+, enquanto o outro empregou o algoritmo embarcado em um aplicativo móvel. Os resultados demonstram a capacidade do sistema de identificar com precisão padrões de condução em evolução, contribuindo para práticas de direção mais seguras, otimização do consumo de combustível e sistemas inteligentes de transporte (ITS, do inglês *Intelligent Transport Systems*). Esta pesquisa avança o campo da *AI* evolutiva ao integrar modelos adaptativos de aprendizado de máquina com soluções embarcadas de *IoT*, permitindo o monitoramento autônomo e auto-organizável do comportamento do motorista.

Palavras-chave: Veículos Inteligentes; *Soft sensors*; Agrupamento Online Incremental; TinyML; Classificação do comportamento do Motorista.

Abstract

Evolving and adaptive intelligent systems are essential for handling dynamic data streams from real-world environments, especially in resource-constrained settings such as vehicles enabled with the Internet of Things (IoT). This study proposes an online evolutionary clustering approach for real-time driver behavior diagnosis, leveraging Tiny Machine Learning (TinyML) and soft sensors. It introduces the MMCloud algorithm, an incremental clustering model designed to manage continuous data streams. The algorithm eliminates the need for retraining and is capable of adapting to concept drift and varying driving conditions. The methodology integrates onboard diagnostics (OBD-II) with an edge computing framework to classify driver behavior into three categories: cautious, normal, and aggressive. To validate the proposed approach, two case studies were conducted in urban environments under various traffic conditions. One used a Freematics One+ device, while the other deployed the embedded algorithm in a mobile application. The results demonstrate the system's ability to accurately identify evolving driving patterns, contributing to safer driving practices, fuel consumption optimization, and intelligent transportation systems (ITS). This research advances the field of evolutionary AI by integrating adaptive machine learning models with embedded IoT solutions, enabling autonomous and self-organizing monitoring of driver behavior.

Keywords: Intelligent Vehicles; Soft sensors; Incremental Online Clustering; TinyML; Driver Behavior Classification.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Abreviaturas

μ	Média dos dados
σ	Desvio padrão dos dados
AOIL	*Adaptive Online Incremental Learning*
ARI	*Adjusted Rand Index*
BGSS	*Between-Group Sum of Squares*
CAN	*Controller Area Network*
CH	Índice Calinski-Harabasz
D	Índice de Dunn
DB	Índice Davies-Bouldin
DBSCAN	(*Density-Based Spatial Clustering of Applications with Noise*
DL	*Deep Learning*
GMM	*Gaussian Mixture Models*
IA	Inteligência Artificial
IoIV	*Internet of Intelligent Vehicles*
IoT	*Internet of Things*
ITS	*Intelligent Transportation Systems*
MAD	*Median Absolute Deviation*
ML	*Machine Learning*
MSM	*Move-Split-Merge*
MSMFCM	*Move-Split-Merge based Fuzzy C-Means*
MTST	*Modified Time Series Transformer*
NMI	*Normalized Mutual Information*

OBD-II *On-Board Diagnostics-II*

PCA *Principal Component Analysis*

PIDs Parâmetros de Identificação de Diagnóstico

PMWFCM *Possibility-based MultiKernel Weighted Fuzzy Clustering Algorithm*

PTQ *Post-Training Quantization*

QP Questões de Pesquisa

RD *Redução de Dimensionalidade*

RPM Rotações por Minuto

SBM *Similarity-Based Modeling*

SIL Índice Silhouette

SLM *Small Language Model*

SQD Soma dos Quadrados das Distâncias

SVM *Support Vector Machine*

t-SNE *T-Distributed Stochastic Neighbor Embedding*

TinyML *Tiny Machine Learning*

VQ *Vector Quantization*

VRC Variance Ratio Criterion

WGSS *Within-Group Sum of Squares*

Seção 1

Introdução

A Internet das Coisas (IoT, do inglês *Internet of Things*) tem impulsionado o setor automotivo ao conectar veículos a uma rede global de dispositivos e sistemas inteligentes, dando origem ao conceito de Internet dos Veículos Inteligentes (IoIV, do inglês *Internet of Intelligent Vehicles*) (Alalwany & Mahgoub 2024, Medeiros et al. 2024). Nesse cenário, um componente central dessa transformação é o sistema de Diagnóstico a Bordo (OBD-II, do inglês *On-board Diagnostics*-II), um protocolo padronizado que permite a extração de diversos dados do veículo — como desempenho do motor, emissões e comportamento do motorista — por meio de *hardwares* compatíveis com esse protocolo, como *scanners* automotivos e módulos embarcados (Kumar & Jain 2023). A análise em tempo real desses dados viabiliza soluções, como a caracterização do comportamento de condução, essencial para promover segurança e eficiência no trânsito (Yen et al. 2021, Neumann 2024).

Contudo, a análise em tempo real de fluxos de dados provenientes de sensores veiculares apresenta desafios técnicos (Soumaya et al. 2017, Amutha et al. 2021). Dentre eles, destaca-se a necessidade de algoritmos de aprendizado de máquina capazes de operar em ambientes com restrições de recursos computacionais, como dispositivos embarcados de baixo consumo energético (Kezia & Anusuya 2023, Gorrab et al. 2024, Mohammadnazar et al. 2024). Deste modo, o aprendizado de máquina compacto (TinyML, do inglês *Tiny Machine Learning*) permite a implementação de algoritmos de aprendizado diretamente na borda da rede, reduzindo a dependência de infraestrutura centralizada e otimizando a latência (Din et al. 2024).

Além disso, a aplicação de algoritmos de aprendizado não supervisionado em fluxos de dados, como os algoritmos incrementais e evolutivos, tem-se mostrado uma abordagem promissora para lidar com mudanças nos padrões dos dados e detectar novos comportamentos sem a necessidade de reprocessamento completo (Fadhil & Sarhan 2020, Da Silva et al. 2020, Gorrab et al. 2024). Essa característica é particularmente relevante em cenários dinâmicos, como o monitoramento do comportamento do motorista, onde novas condições de tráfego e estilos de direção surgem continuamente (Guan et al. 2022, Bouhsissin et al. 2023).

Motivado pelas lacunas identificadas na literatura — como a necessidade de algoritmos capazes de lidar com mudanças nos padrões de dados em fluxos contínuos, a escassez de soluções não supervisionadas adaptativas para o monitoramento de comportamento veicular e os desafios associados a ambientes embarcados com restrições computacionais e energéticas —, este estudo busca avançar no enfrentamento dessas limitações com uma

abordagem orientada ao contexto veicular.

Neste trabalho, é proposto o algoritmo MMCloud, um modelo de *clustering* não supervisionado online, integrado a *soft sensors* e à rotulação de dados, com o objetivo de classificar e analisar o comportamento do motorista. Os *soft sensors*, ou sensores virtuais, correspondem a variáveis que não estão fisicamente presentes no veículo, mas que podem ser calculadas ou estimadas a partir de modelos matemáticos ou heurísticas (Andrade et al. 2022). O MMCloud combina técnicas de aprendizado incremental com métodos de análise de dados típicos e excêntricos, permitindo a adaptação contínua do sistema às mudanças nos padrões de direção e nas condições dinâmicas do tráfego. Esse fenômeno, conhecido como desvio de conceito (do inglês, *concept drift*), refere-se à alteração gradual ou abrupta na distribuição estatística dos dados ao longo do tempo, o que pode comprometer a validade de modelos previamente aprendidos (Hinder et al. 2023). A proposta busca ainda garantir desempenho eficiente em dispositivos embarcados com recursos computacionais e energéticos limitados.

Como estudo de caso, a abordagem proposta foi validada por meio de dois estudos, ambos conduzidos em condições reais na cidade de Natal-RN, Brasil.O primeiro estudo implementou o algoritmo em um dispositivo Freematics One+, que possui o protocolo OBD-II integrado. A validação foi realizada através de um delineamento de validação cruzada motorista-veículo, no qual dois motoristas alternaram a condução de dois veículos distintos. Esta abordagem permitiu a observação e análise do comportamento de direção em múltiplas configurações.

O segundo estudo de caso utilizou uma abordagem mais acessível, integrando o algoritmo a um aplicativo de smartphone conectado via Bluetooth a um dispositivo OBD-II ELM327. Este estudo envolveu seis diferentes pares de motorista-veículo. Para permitir uma análise detalhada dos padrões de direção, cada um desses seis motoristas percorreu a mesma rota específica quatro vezes separadas em seus respectivos veículos. Os dados foram coletados em horário comercial ao longo de uma semana. Deste modo, destaca-se que ambos os estudos compartilharam a mesma rota, com características variadas como rotatórias, semáforos e diferentes tipos de curvas, para capturar padrões de direção diversos. Os resultados obtidos em ambos os cenários demonstraram a capacidade da abordagem de identificar padrões de comportamento de direção, oferecendo *insights* para melhorar a segurança e eficiência veicular.

Esta proposta busca contribuir para o avanço dos Sistemas de Transporte Inteligentes (ITS, do inglês *Intelligent Transportation Systems*), integrando conceitos de IoT, *Edge Computing* e aprendizado de máquina para desenvolver uma solução sustentável e adaptativa para a análise do comportamento de motoristas (Alalwany & Mahgoub 2024, Medeiros et al. 2024).

1.1 Objetivos

O objetivo geral deste trabalho é investigar a abordagem de monitoramento e diagnóstico do comportamento do motorista em tempo real, visando melhorar a segurança veicular, otimizar o consumo de combustível e promover práticas de direção mais eficientes e seguras. Para alcançar esse objetivo, este estudo se concentra nos seguintes objetivos

específicos:

1. Investigar a utilização de *soft sensors* e técnicas de TinyML para processar os dados de forma eficiente em dispositivos embarcados, permitindo a análise em tempo real e a adaptação do sistema a diferentes padrões de direção e condições de tráfego dinâmicas.

2. Desenvolver e avaliar a aplicação do algoritmo MMCloud para análise e classificação do comportamento do motorista, com foco na detecção de padrões de direção e na identificação de comportamentos de risco, a fim de fornecer diagnósticos em tempo real.

3. Embarcar a solução proposta (algoritmo MMCloud e *soft sensor*) em um ambiente de TinyML, representado por dispositivos de baixo poder de processamento (como Freematics One+ e um aplicativo móvel), garantindo sua implementação eficiente e avaliando sua performance computacional.

4. Analisar o impacto do diagnóstico em tempo real do comportamento do motorista na melhoria da segurança veicular, na redução do consumo de combustível e na adaptação do comportamento do motorista por meio de *feedback* contínuo, tanto preventivo quanto educativo, incentivando a adoção de hábitos de direção mais seguros e eficientes.

5. Validar a solução proposta em condições reais, utilizando uma rota em cenário urbano, para avaliar a precisão do sistema na classificação do comportamento do motorista.

1.2 Justificativa

A necessidade de avanços em sistemas de monitoramento veicular em tempo real é evidente diante da crescente complexidade e conectividade dos veículos modernos (Flores et al. 2023), especialmente em relação ao consumo de combustível e à emissão de CO_2 na atmosfera. Nesse contexto, a aplicação de Inteligência Artificial (IA) e aprendizado de máquina surge como uma resposta prática aos desafios enfrentados pelos setores automotivo e de IoT (Medeiros et al. 2023).

A integração de *soft sensors* e o desenvolvimento de algoritmos *online*, como o MMCloud, em um *hardware* de baixo consumo energético, refletem uma solução eficiente e acessível. Essa abordagem atende às demandas do setor automotivo por sistemas mais inteligentes e conectados, ao mesmo tempo em que aborda preocupações com eficiência energética e sustentabilidade (Signoretti et al. 2021, Silva et al. 2023).

O foco na classificação do comportamento do motorista e na utilização de técnicas de TinyML busca personalizar e adaptar sistemas de monitoramento (Silva 2022). A identificação de padrões de comportamento do condutor contribui para a segurança veicular e possibilita diagnósticos mais precisos e intervenções adaptadas, promovendo uma condução mais segura e eficiente (Lattanzi & Freschi 2021).

Assim, a aplicação dessa solução na IoIV representa uma contribuição tecnológica para esse setor em crescimento. A criação de uma abordagem capaz de realizar análises em tempo real e rotular o padrão de condução do motorista, alinhada aos princípios da IoIV, diferencia essa abordagem proposta de outras iniciativas.

1.3 Metodologia da Pesquisa

Para atingir tais objetivos, foi necessário definir a classificação quanto ao método de pesquisa. Nesse contexto, em relação à natureza da pesquisa, ela foi caracterizada como aplicada, pois tem o propósito de gerar conhecimentos aplicáveis para solucionar um problema específico (Dresch et al. 2020, Wazlawick 2020).

Quanto à abordagem dos dados, a pesquisa foi classificada como qualitativa, uma vez que as variáveis estavam associadas à qualidade da classificação (Wazlawick 2020). No que diz respeito aos objetivos, o estudo foi caracterizado como explicativo, conforme proposto por Dresch et al. (2020) e Gil (2022), visando identificar os fatores primordiais para a ocorrência de um fenômeno, além de estabelecer uma hipótese de causa e efeito em relação ao fenômeno estudado.

Por outro lado, em relação aos procedimentos técnicos, a pesquisa foi definida como um estudo de caso, buscando uma compreensão mais abrangente e com maior foco na visão de mundo dos setores populares, de forma mais objetiva e com validade prática através de um estudo de caso (Dresch et al. 2020, Wazlawick 2020).

1.4 Estrutura da Dissertação

Este documento está organizado da seguinte maneira. Nesta seção, discutiu-se a motivação principal para o trabalho proposto. Desse modo, foram apresentadas a contextualização da pesquisa, problemática e hipóteses, objetivos e justificativa, os quais fornecem um conhecimento geral sobre o contexto de aplicação desta pesquisa. A Seção 2 explana os conceitos e termos necessários para o entendimento do trabalho. Na Seção 3, apresenta-se e discute-se os trabalhos relacionados a esta pesquisa, ressaltando seus pontos fortes e as lacunas que podem ser exploradas e preenchidas. Na Seção 4, apresenta-se a metodologia proposta por este estudo, enquanto que a Seção 5 detalha o planejamento, a operação e a coleta de dados dos estudos de caso realizados para validar a abordagem proposta. A Seção 6 apresenta alguns resultados obtidos a partir dos estudos de caso. Por fim, na Seção 7, são discutidas as conclusões iniciais, bem como serão apresentados os artigos publicados e submetidos.

Seção 2

Fundamentação Teórica

Nesta seção, são apresentados os principais conceitos teóricos necessários para o entendimento da solução desenvolvida ao longo deste trabalho. Assim sendo, será discutido o que é aprendizado de máquina e TinyML, em particular, focando na parte de aprendizagem online não supervisionada e nas suas principais métricas de avaliação. Por fim, será explorado o *soft-sensor* do gráfico de radar utilizado na abordagem proposta.

2.1 *Aprendizado de Máquina*

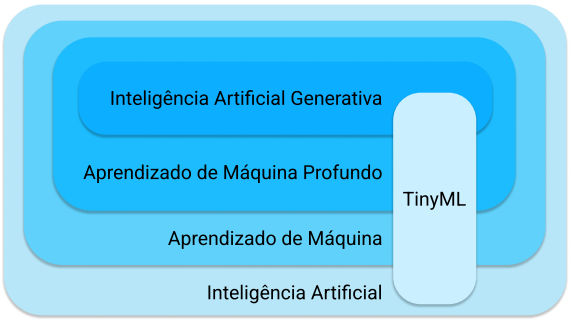
O Aprendizado de Máquina (ML, do inglês *Machine Learning*) é uma subdivisão da IA, conforme observa-se na Figura 2.1, que permite às máquinas aprender a partir de dados e melhorarem seu desempenho ao longo do tempo, sem intervenção humana direta (Mitchell 1997). Isso é alcançado por meio de algoritmos e modelos que identificam padrões nos dados, fazem previsões e tomam decisões com base nessas informações (Géron 2022).

Dentro do campo de ML, existem áreas como o Aprendizado Profundo (DL, do inglês *Deep Learning*), que utiliza redes neurais com múltiplas camadas para aprender representações complexas de dados, sendo fundamental para tarefas como reconhecimento de imagens e processamento de linguagem natural (Géron 2022). Além disso, a IA Generativa é um campo da IA que se concentra na criação de novos dados, como imagens e textos, que se assemelham aos dados de treinamento, mas são originais (Raschka 2024).

É possível ainda dividir as técnicas de ML em algumas categorias, como: aprendizado supervisionado, não supervisionado, semi-supervisionado e por reforço. No aprendizado supervisionado, os algoritmos são treinados em um conjunto de dados rotulados, o que significa que o modelo aprende a associar entradas a saídas específicas. No aprendizado não supervisionado, os algoritmos exploram padrões nos dados sem o benefício de rótulos explícitos, identificando agrupamentos ou estruturas subjacentes (Géron 2022, Huyen 2022).

Na Figura 2.2, observa-se que ainda existem algumas subdivisões dentro de cada tipo de aprendizado. Por exemplo, no aprendizado supervisionado é possível ter uma tarefa de classificação, em que se está querendo prever uma determinada categoria para uma instância, e de regressão, em que se está querendo prever um número a partir das características da instância. Já no aprendizado não supervisionado, é possível ter a tarefa de redução

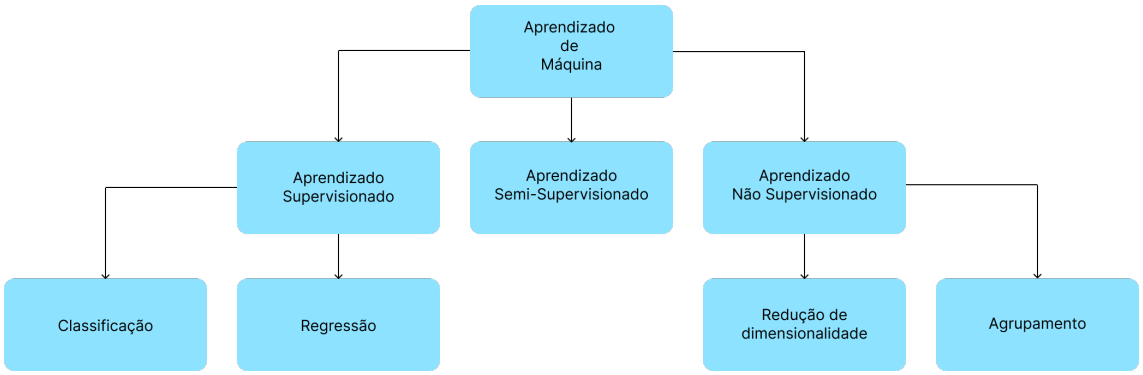
Figura 2.1: Diagrama de alto nível das sub-áreas da inteligência artificial.



Fonte: Adaptado de Raschka (2024).

de dimensionalidade (RD), em que deseja-se obter as *features* com maior representatividade do conjunto de dados, ou de agrupamento, em que se deseja agrupar as instâncias de dados com base em suas características.

Figura 2.2: Tipos de aprendizado de máquina.



Fonte: Elaborado pelo Autor, 2025.

Na aprendizagem semi-supervisionada, o algoritmo busca rotular os dados com base nos exemplos rotulados que ele dispõe ou a partir de grupos detectados pela parte não supervisionada. Nesse sentido, os algoritmos desse tipo podem tanto ser compostos por apenas um modelo que é capaz de lidar com os dados rotulados e não rotulados, quanto pode ser composto por dois modelos, um não supervisionado para realizar o agrupamento dos dados e outro supervisionado para realizar a atribuição do rótulo a partir do grupo. Por fim, o aprendizado por reforço envolve agentes que interagem com um ambiente e aprendem a tomar decisões que maximizam uma recompensa ao longo do tempo (Géron 2022).

No contexto de IoT, o aprendizado de máquina desempenha um papel importante na análise em tempo real dos dados gerados por sensores e dispositivos interconectados (Yang & Shami 2022). Essa característica faz com que a natureza dos dados seja não rotulada, ou seja, os dados chegam através de um fluxo e o algoritmo precisa reconhecer o padrão do dado atual antes que o próximo seja recebido (Ashfahani & Pratama 2022). Como a natureza do problema discutido neste trabalho encontra-se nesse contexto, as

categorias de problemas da aprendizagem não supervisionada serão discutidas de forma mais detalhada na próxima seção.

2.1.1 Aprendizado não supervisionado

O Aprendizado não supervisionado tem como objetivo a descoberta de padrões, a identificação de relações intrínsecas e a revelação de estruturas ocultas em dados não rotulados, não classificados e não categorizados, ficando a cargo do usuário a interpretação da saída do algoritmo (Géron 2022). Em outras palavras, ele encontra agrupamentos naturais e, de formal geral, mutuamente exclusivos nos dados, baseados na similaridade entre as instâncias (Patel 2019).

Em linguagem matemática, o aprendizado não supervisionado consiste em examinar múltiplas ocorrências de uma variável aleatória X — que pode ser representada por um escalar, vetor ou estrutura mais complexa — e adquirir conhecimento sobre a distribuição de probabilidade $p(X)$ relativa a essas ocorrências (Ashenden 2021, Verdhan 2025). Existem quatro técnicas principais dentro da aprendizagem não supervisionada:

1. Agrupamento (*Clustering*): O agrupamento envolve a identificação de padrões que sugerem a existência de grupos ou *clusters* de dados semelhantes. Os algoritmos de agrupamento, como o K-Means e o *Hierarchical Clustering*, procuram agrupar os pontos de dados de acordo com a similaridade entre eles. Isso é fundamental em diversas aplicações, como segmentação de clientes, análise de redes sociais e classificação de documentos (Géron 2022).

2. Detecção de *Outliers*: A detecção de anomalias busca identificar padrões incomuns ou desvios significativos em dados não rotulados. Nesse contexto, o objetivo é distinguir observações que diferem substancialmente da maioria, com base em características estatísticas ou estruturais dos dados. Essa técnica é amplamente aplicada em áreas como detecção de fraudes, identificação de falhas em processos industriais e limpeza de dados para remoção de *outliers* (Géron 2022).

3. Redução de Dimensionalidade: Redução de dimensionalidade refere-se à técnica de reduzir a complexidade dos dados, mantendo as informações essenciais. Isso é feito reduzindo o número de variáveis ou dimensões nos dados originais. A Análise de Componentes Principais (PCA, do inglês, *Principal Component Analysis*) e o *T-Distributed Stochastic Neighbor Embedding* (t-SNE) são exemplos de métodos que permitem a visualização de dados em dimensões reduzidas, mantendo a estrutura dos dados originais (Tripathy et al. 2021, Géron 2022).

4. Regras de associação: As regras de associação é uma técnica voltada para a descoberta de relações entre atributos em grandes conjuntos de dados. Essa abordagem é frequentemente aplicada em contextos como análise de vendas em supermercados, onde o objetivo é identificar padrões de compra, como a associação entre a compra de *barbecue sauce* e batatas *chips* com a compra de carne. Ela permite extrair *insights* como a disposição estratégica de produtos para aumentar as vendas, por exemplo (Géron 2022).

Como este estudo utiliza as técnicas de agrupamento e de detecção de *outliers*, as duas próximas sub-seções serão focadas em seus respectivos detalhamentos.

2.1.2 Clusterização

Clusterização é uma técnica de aprendizado não supervisionado que organiza dados em grupos, denominados *clusters*, com base em características semelhantes (Bonaccorso 2019, Géron 2022). Dados pertencentes ao mesmo *cluster* possuem maior similaridade entre si do que com dados de *clusters* diferentes. A clusterização é empregada em diversas áreas, como segmentação de mercado, análise de dados biológicos, processamento de imagens e redes sociais (Ashenden 2021).

Formalmente, a clusterização consiste em particionar dados em um espaço R^n em k *clusters*. De forma geral, um ponto x_i é atribuído a um *cluster* C_j com base em um critério de distância ou similaridade. No algoritmo K-Means, por exemplo, minimiza-se a soma das distâncias quadradas entre os pontos de um *cluster* e seu centróide μ_j:

$$\sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2,$$

onde μ_j é o centróide do *cluster* C_j, calculado pela média dos pontos associados (Gan et al. 2020).

Tipos de Algoritmos de Clusterização

Os algoritmos de clusterização podem ser organizados em diferentes tipos, de acordo com a abordagem adotada. A seguir, apresentam-se alguns tipos e exemplos comuns:

1. Algoritmos Particionais: dividem os dados em um número fixo de *clusters*, como no K-Means, que requer a definição prévia do número de *clusters* (k) e busca minimizar a variabilidade interna (Géron 2022).

2. Algoritmos Hierárquicos: criam uma hierarquia de *clusters* por meio de abordagens aglomerativas ou divisivas, úteis para explorar estruturas em conjuntos de dados (Patel 2019).

3. Algoritmos Baseados em Densidade: identificam *clusters* com base na densidade de pontos, como o DBSCAN, que lida bem com formas complexas e ruídos (Patel 2019).

4. Algoritmos Baseados em Modelos: assumem que os dados foram gerados por uma combinação de modelos probabilísticos, como nos Modelos de Mistura Gaussiana (GMM), que representam *clusters* como distribuições gaussianas (Géron 2022).

A clusterização é aplicada em diferentes contextos, como na segmentação de clientes, onde consumidores são agrupados de acordo com padrões de compra, permitindo estratégias personalizadas de marketing; na análise genômica, para identificar padrões em dados biológicos; no processamento de imagens, para separar regiões com propriedades similares e facilitar análises visuais; e na análise de redes sociais, para identificar grupos de interação com interesses comuns (Patel 2019, Géron 2022).

Um dos principais desafios encontrados por essa técnica é a necessidade de determinar o número de *clusters* (k) a priori em métodos como o K-Means, o que pode introduzir subjetividade ou exigir métodos adicionais para avaliação, como o critério do cotovelo ou a silhueta (Gan et al. 2020, Géron 2022). Além disso, a sensibilidade a escalas e unidades das variáveis pode impactar os resultados, tornando necessária a normalização dos dados (Gan et al. 2020). Por fim, também é importante considerar a escalabilidade dos algoritmos, uma vez que métodos como a clusterização hierárquica possuem alta complexidade computacional, dificultando a aplicação em grandes volumes de dados (Gan et al. 2020, Géron 2022).

2.1.3 Detecção de Outliers

A detecção de *outliers* é uma etapa fundamental em diversas aplicações de aprendizado de máquina e análise de dados, pois esses valores podem distorcer análises estatísticas e impactar negativamente o desempenho de algoritmos de aprendizado (Cesarone et al. 2024). Um *outlier* pode ser definido como uma observação que se desvia significativamente de outras observações no conjunto de dados, geralmente devido a erros de medição, ruído, ou comportamentos incomuns (Kennedy 2025).

Uma abordagem comum para detecção de *outliers* é o uso da média e do desvio padrão dos dados (Montgomery & Runger 2020). Suponha que tenhamos um conjunto de dados $X = \{x_1, x_2, \ldots, x_n\}$, com média μ e desvio padrão σ. Um ponto x_i é considerado um *outlier* se satisfizer a condição:

$$|x_i - \mu| > k \cdot \sigma,$$

onde k é um fator de escala que determina a sensibilidade do critério. Valores comuns para k incluem 2, 2,54 ou 3, dependendo do nível de confiança desejado.

O critério de desvio padrão pode ser associado a um nível de confiança com base na distribuição normal dos dados (Montgomery & Runger 2020). Por exemplo:

- Para $k = 2$, aproximadamente 95,45% dos dados caem dentro de $2 \cdot \sigma$ da média, o que implica uma confiança de 4,55% de que um valor fora desse intervalo é um *outlier*.
- Para $k = 2,54$, a confiança de que um valor fora do intervalo seja um *outlier* aumenta para cerca de 98,89%.
- Para $k = 3$, a confiança aumenta ainda mais, para aproximadamente 99,73%.

No contexto do algoritmo proposto, a detecção de *outliers* baseada na média e no desvio padrão é particularmente útil devido à sua simplicidade computacional e adequação a ambientes com restrições de recursos. Além disso, a interpretação do resultado é direta, permitindo identificar e tratar pontos atípicos de forma eficiente.

2.2 *Tiny Machine Learning* e Computação de Borda

De acordo com Iodice & Naughton (2022), TinyML abrange tecnologias de aprendizado de máquina e sistemas embarcados que viabilizam aplicações inteligentes em dis-

positivos com restrições energéticas e de *hardware*. Nessas aplicações, a limitação de energia disponível — comum em dispositivos alimentados por bateria ou em contextos com infraestrutura restrita — impõe a adoção de *hardware* de baixo consumo energético, o que, por sua vez, restringe o poder computacional disponível. Apesar dessas limitações, esses dispositivos são equipados com sensores capazes de captar informações do ambiente físico e tomar decisões utilizando algoritmos de aprendizado de máquina. Essa combinação torna o TinyML uma solução promissora para cenários em que eficiência energética e processamento local são indispensáveis.

Para que os sistemas TinyML funcionem adequadamente, é essencial alinhar os algoritmos de ML às características específicas dos dispositivos onde serão implementados. Conforme destacado por Flores et al. (2023), a compreensão das restrições de *hardware* é importante para o desenvolvimento de arquiteturas eficazes. Dispositivos IoT, que frequentemente utilizam microcontroladores com consumo energético inferior a 1 mW, memória RAM limitada a algumas centenas de *kilobytes* e velocidades de *clock* de poucas dezenas de *megahertz*, exemplificam bem essas limitações (Warden & Situnayake 2019). A integração eficiente entre *software* e *hardware* é, portanto, uma etapa fundamental nesse processo.

Essa integração ganha ainda mais relevância no contexto da Computação de Borda (*Edge Computing*), que processa dados localmente em dispositivos próximos à fonte geradora. Em vez de depender de um *data center* remoto, essa abordagem visa reduzir a latência, melhorar o uso da largura de banda e aumentar a eficiência geral do sistema (Cao et al. 2020). Assim, a Computação de Borda se apresenta como um paradigma complementar ao TinyML, atendendo a demandas crescentes por respostas rápidas e descentralizadas.

Quando combinados, os princípios de Computação de borda e do TinyML permitem a análise de dados e a tomada de decisões em tempo real, mesmo em ambientes onde a latência é crítica (Situnayake & Plunkett 2023). Essa sinergia expande as possibilidades de uso de dispositivos IoT, que passam a ser capazes de processar informações na borda da rede de forma eficiente. Com isso, abrem-se oportunidades para aplicações em áreas diversas, como monitoramento ambiental, automação industrial e saúde, consolidando o potencial dessa abordagem para cenários dinâmicos e distribuídos.

2.3 Aprendizado Online e Incremental

O aprendizado online, também conhecido como aprendizado contínuo ou *lifelong*, permite que um modelo de aprendizado de máquina atualize seus parâmetros conforme novos dados são disponibilizados, sem a necessidade de repetir o treinamento completo (Zhang et al. 2021, Hoi et al. 2021). Essa abordagem é útil em cenários onde os dados chegam de forma sequencial, como fluxos de dados em tempo real, e onde o armazenamento e o processamento de grandes volumes de dados simultaneamente são inviáveis (Yan et al. 2025). Assim, o aprendizado online se apresenta como uma solução eficiente para lidar com ambientes que demandam flexibilidade e adaptabilidade constantes.

O aprendizado incremental, por outro lado, expande os princípios do aprendizado online ao incorporar novos dados sem comprometer o conhecimento previamente adquirido,

evitando o problema conhecido como *catastrophic forgetting* (Parisi et al. 2019). Esse método equilibra a incorporação de novas informações e a preservação do histórico, garantindo que o modelo mantenha sua performance em cenários com mudanças graduais ou recorrentes (Zhang et al. 2021, Parisi et al. 2019). Dessa forma, enquanto o aprendizado online prioriza a atualização contínua, o aprendizado incremental assegura a manutenção do aprendizado acumulado.

Formalmente, o aprendizado incremental pode ser descrito considerando um conjunto de dados sequenciais $x_1, x_2, \ldots, x_t$. Partindo de um modelo existente no instante $t-1$, M_{t-1}, o objetivo é ajustá-lo para obter um modelo atualizado, M_t, utilizando apenas o dado atual, x_t, e informações auxiliares, como estatísticas acumuladas (Hoi et al. 2021). Essa formulação demonstra como o aprendizado incremental é projetado para operar com recursos limitados, sem a necessidade de reprocessar dados antigos.

Ao combinar as capacidades do aprendizado online e incremental, essa abordagem torna-se uma escolha apropriada para aplicações que exigem respostas em tempo real e operam sob restrições computacionais. Dispositivos embarcados e sistemas de IoT exemplificam bem esse cenário, beneficiando-se da eficiência energética e da capacidade de adaptação que essas técnicas oferecem (Ren et al. 2021). Assim, o aprendizado online incremental se consolida como uma solução prática e robusta para desafios contemporâneos no processamento de dados em tempo real.

Seção 3

Trabalhos Relacionados

A crescente geração de fluxos de dados em tempo real e a necessidade de processá-los de maneira eficiente têm contribuído para desafios no desenvolvimento de algoritmos de *clustering* online. Esses desafios são ainda mais intensificados quando é necessário lidar com mudanças nos padrões dos dados, conhecidos como *concept drift*, e garantir que os algoritmos operem de forma eficiente em dispositivos com recursos limitados, como sistemas embarcados. Embora diversas abordagens tenham sido propostas para resolver problemas de aprendizado incremental em fluxos de dados contínuos, existem lacunas importantes que ainda precisam ser abordadas, principalmente no que diz respeito à eficiência computacional e à aplicabilidade em cenários específicos, como o monitoramento do comportamento de motoristas em tempo real.

O presente estudo consiste em uma continuação dos trabalhos de Silva (2022) e Medeiros et al. (2024). O trabalho de Silva (2022) propôs uma metodologia abrangente, orientada a fluxos de dados, para modelagem do comportamento do motorista utilizando algoritmos de aprendizado não supervisionado online, adaptando especificamente os conceitos de TEDA e AutoCloud. Esse estudo estabeleceu a viabilidade de detectar eventos de direção e modelar o comportamento por meio de uma abordagem evolutiva e autônoma, sem a necessidade de treinamento supervisionado. Com base nisso, Medeiros et al. (2024) apresentaram explorações preliminares na aplicação de algoritmos não supervisionados adaptativos para TinyML no contexto do comportamento do motorista. Nesse sentido, este estudo estende diretamente esses conceitos fundamentais, introduzindo e validando o algoritmo MMCloud, projetado para classificação do comportamento do motorista em tempo real, em dispositivos, dentro do paradigma TinyML. O MMCloud se distingue por sua natureza incremental, adaptação dinâmica a *concept drift* através de limiares de *outliers* e divisão de *clusters* evolutivos, e eficiente computacionalmente de forma a executar em *hardwares* limitados computacionalmente. Essa combinação de conceitos ainda não foi amplamente explorada na literatura.

Zhang et al. (2021) apresentaram o *Adaptive Online Incremental Learning* (AOIL), um modelo que aborda os desafios de *concept drift* e *catastrophic forgetting*, problemas recorrentes em fluxos de dados não estacionários. O AOIL combina *autoencoders* com módulos de memória para capturar representações latentes robustas e detectar mudanças nos padrões de dados. O algoritmo utiliza uma perda de reconstrução para identificar variações na distribuição de probabilidade dos dados e mecanismos de atenção para fusão de características latentes. Os resultados experimentais demonstraram melhorias

significativas na robustez e na acurácia em comparação com outros métodos do estado da arte. Entretanto, o AOIL não foi projetado para ambientes de restrição de recursos, como dispositivos embarcados, nem para aplicação em classificações específicas, como comportamento de motoristas.

Nordahl et al. (2022) apresentaram o *EvolveCluster*, uma abordagem que utiliza segmentação de dados para capturar tendências temporais em fluxos de dados contínuos. O algoritmo integra novos segmentos de dados sem reprocessamento completo, garantindo uma solução eficiente. O trabalho realiza a avaliação do método proposto em vários conjuntos de dados de *benchmark*, mas não explora contextos específicos como o comportamento do motorista. Além disso, os autores realizaram a análise da complexidade computacional do algoritmo, porém eles não mencionam se o algoritmo foi, de fato, embarcado em um *hardware* com limitações computacionais.

Abernathy & Celebi (2022) propuseram uma extensão incremental do algoritmo *K-Means* para *clustering* online, com foco em eficiência computacional. Este método suprime a dependência de inicializações explícitas de centros de *clusters* e processa cada ponto de dados apenas uma vez, reduzindo significativamente a complexidade computacional. Os resultados mostraram que o algoritmo alcança uma qualidade de *clustering* próxima à do K-Means tradicional, mas com maior eficiência. Contudo, o trabalho não aborda fluxos de dados evolutivos nem considera as limitações de dispositivos com recursos restritos.

Angelova et al. (2024) desenvolveram o *EdgeCluster*, um algoritmo de *clustering* evolutivo projetado para dispositivos de borda com capacidade limitada de armazenamento e processamento. O algoritmo mantém *clusters* atualizados dinamicamente, adaptando-se a mudanças nos fluxos de dados enquanto monitora padrões anômalos em tempo real. Apesar de ser projetado para dispositivos de borda, o estudo não analisou o desempenho do algoritmo proposto em um dispositivo dessa natureza. Além disso, o estudo não considera a aplicação em cenários específicos, como a classificação de comportamento de motoristas.

Gorrab et al. (2024) introduziram o *Split Incremental Clustering Algorithm*, que aplica uma abordagem baseada em divisão dinâmica para dados mistos. O algoritmo gerencia a chegada contínua de novos dados e adapta a estrutura dos *clusters* existentes em resposta a mudanças nos atributos ou na distribuição de classes. Os experimentos indicaram uma melhora no tempo de processamento e na qualidade do *clustering* em cenários de dados heterogêneos. Apesar disso, o trabalho não explora a aplicação do modelo em dispositivos embarcados nem considera cenários específicos como o comportamento de motoristas.

Ba & Gu (2024) apresentaram o *Move-Split-Merge based Fuzzy C-Means* (MSMFCM), um algoritmo para *clustering* de séries temporais ruidosas. O método utiliza funções *wavelet* dinâmicas otimizadas por *Median Absolute Deviation* (MAD) para reduzir ruídos, uma métrica de distância *Move-Split-Merge* (MSM) para avaliar similaridade, e inicialização *K-Means++* no *Fuzzy C-Means* para evitar mínimos locais. Em experimentos com 20 conjuntos de dados, o MSMFCM apresentou ganhos médios de 26,09% no *Adjusted Rand Index* (ARI) e 18,86% no *Normalized Mutual Information* (NMI), superando métodos como *K-Means* e *K-Medoids*. Contudo, o algoritmo não considera fluxos contínuos nem foi avaliado em dispositivos embarcados ou cenários como a classificação de

comportamento de motoristas.

Almeida et al. (2025) apresentaram um modelo evolutivo baseado no *Similarity-Based Modeling* (SBM) para *clustering* online em ambientes não estacionários. O algoritmo utiliza uma função de pertencimento que considera o erro de aproximação e a densidade dos dados, possibilitando a modelagem de *clusters* de formas arbitrárias com baixo esforço computacional. Os experimentos mostraram resultados superiores em métricas como ARI e Pureza, comparados a métodos do estado da arte. Contudo, o trabalho não aborda restrições computacionais de dispositivos embarcados, nem foi aplicado a classificações específicas, como comportamento de motoristas.

No domínio da classificação do comportamento de motoristas, Yarlagadda et al. (2021) desenvolveram um *framework* para detectar padrões agressivos de direção em veículos pesados de passageiros. Para classificar milhares de eventos de aceleração e frenagem extraídos, empregaram aprendizado de máquina não supervisionado, com Análise de Componentes Principais (PCA) para redução de dimensionalidade e *clustering* K-Means em duas etapas. Os resultados indicaram que 86,5% das acelerações e 65,3% das frenagens foram não agressivas. Apesar da coleta dos dados ter sido realizada em tempo real, a análise para definição e caracterização dos padrões é realizada de maneira offline sobre o conjunto de dados completo. Além disso, o trabalho não aborda as restrições computacionais para uma eventual implementação do modelo de classificação em dispositivos embarcados para operação em tempo real.

Govers et al. (2023) focaram na identificação do motorista usando dados multivariados de séries temporais de sensores de veículos, propondo modelos BiLSTM-A e Modified Time Series Transformer (MTST). Embora alcançando alta precisão, especialmente com conjuntos de séries temporais com deslocamento, sua abordagem é voltada para identificar quem está dirigindo em vez de classificar como eles estão dirigindo de forma não supervisionada e evolutiva, e seus modelos foram avaliados em *hardware* de servidor, não explicitamente para contextos TinyML.

Valente et al. (2024) descreveram o desenvolvimento do *DriverAlert*, um aplicativo móvel que coleta dados de sensores de celulares e APIs públicas para classificar estilos de direção. Foram testados modelos como K-Means, *Random Forest* e *Support Vector Machine* (SVM) para identificar perfis de motoristas. O estudo destaca a coleta integrada de dados e a análise em uma plataforma de visualização. No entanto, o modelo não considera fluxos contínuos de dados nem a aplicação em dispositivos com restrição de recursos.

Xie et al. (2024) investigaram comportamentos de condução de alta emissão de veículos diesel pesados (HDDV) considerando o grau da estrada e os princípios do motor. Eles utilizaram dados OBD e aprendizado de máquina (*Random Forest*, XGBoost, *Elastic Net*) para estimar emissões de NOx e CO2 e identificar características relacionadas ao motor ligadas a altas emissões, como abertura do pedal do acelerador e carga do motor. Embora valioso para entender os impulsionadores de emissões, este trabalho foca na previsão e explicação de emissões, e não em *clustering* online e evolutivo de comportamento para aplicações TinyML.

Ma et al. (2025) apresentaram o *Possibility-based MultiKernel Weighted Fuzzy Clustering Algorithm* (PMWFCM). Este modelo combina múltiplos *kernels* para lidar com

ruídos e *outliers*, características comuns em dados de tráfego. Os resultados destacam melhorias na robustez e na identificação de padrões complexos em comportamentos de condução. No entanto, o algoritmo é mais adequado para processamento offline e não foi testado em dispositivos embarcados ou com fluxos de dados em tempo real.

Fang et al. (2025) propuseram uma estrutura para reconhecimento do comportamento de condução de HDV e avaliação da economia de combustível usando o algoritmo *Toeplitz Inverse Covariance Clustering* (TICC) para segmentação e *clustering* offline de dados históricos, com uma fase de serviço online para mapear comportamentos em tempo real para esses *clusters* pré-treinados. Embora abordando o reconhecimento de comportamento e o *feedback* de condução ecológica, essa abordagem depende do treinamento de modelos offline e não apresenta um mecanismo de *clustering* evolutivo e incremental diretamente no dispositivo de borda.

Para facilitar a compreensão, a Tabela 3.1 organiza os trabalhos analisados de acordo com três categorias principais: *clustering online incremental*, *classificação do comportamento do motorista* e *TinyML*.

Tabela 3.1: Classificação dos trabalhos relacionados por categorias.

Características / Trabalho	Clustering Online Incremental	Driver Behavior	TinyML
Zhang et al. (2021)	✓	✖	✖
Nordahl et al. (2022)	✓	✖	✖
Abernathy & Celebi (2022)	✓	✖	✖
Angelova et al. (2024)	✓	✖	✖
Gorrab et al. (2024)	✓	✖	✖
Ba & Gu (2024)	✓	✖	✖
Almeida et al. (2025)	✓	✖	✖
Yarlagadda et al. (2021)	✖	✓	✖
Govers et al. (2023)	✖	✓	✖
Valente et al. (2024)	✖	✓	✖
Xie et al. (2024)	✖	✓	✖
Ma et al. (2025)	✖	✓	✖
Fang et al. (2025)	✖	✓	✖
Silva (2022)	✓	✓	✖
Medeiros et al. (2024)	✓	✓	✓
Este Estudo (MMCloud)	✓	✓	✓

Fonte: Elaborado pelo Autor (2025).

As contribuições destacadas representam avanços significativos no aprendizado incremental, *clustering* e classificação em tempo real. No entanto, ainda existem lacunas na integração dessas abordagens em dispositivos embarcados de baixo consumo energético, bem como na aplicação em cenários específicos, como a classificação de comportamento de motoristas. Este trabalho busca preencher essa lacuna na área de monitoramento e diagnóstico do comportamento de motoristas, propondo uma solução de *clustering* incre-

mental online adaptada para o ambiente de TinyML. O foco principal é viabilizar o monitoramento e a classificação do comportamento do motorista em tempo real, utilizando um modelo computacional que seja capaz de operar em sistemas embarcados com recursos limitados. Essa abordagem visa não apenas melhorar a segurança veicular e otimizar o consumo de combustível, mas também fornecer *insights* para o desenvolvimento de sistemas de assistência ao motorista que exigem baixa latência e eficiência computacional, características essenciais em ambientes de TinyML.

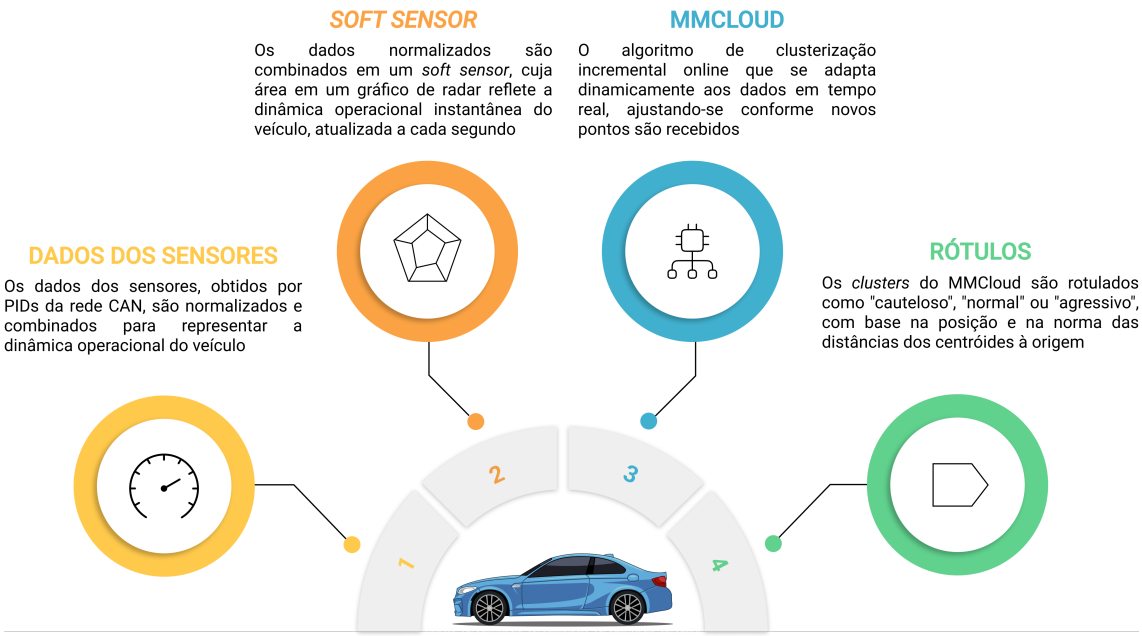
Seção 4

Abordagem Proposta

Esta seção apresenta a abordagem proposta para o processamento de dados veiculares em tempo real para classificar o comportamento do motorista a partir da fusão de *soft sensors* com sensores físicos e a utilização do algoritmo MMCloud.

A Figura 4.1 ilustra a visão geral da abordagem proposta. Na primeira etapa, os dados lidos dos sensores físicos do veículo, conhecidos como Parâmetros de Identificação de Diagnóstico (PIDs), são combinados em um *soft sensor* conhecido como área do radar na segunda etapa. Os valores resultantes desta etapa são, então, utilizados como entrada para a próxima, onde o algoritmo MMCloud irá filtrar valores *outliers* e agrupar os dados. Na quarta etapa, o algoritmo retorna como saída o rótulo do comportamento do motorista (cauteloso, normal ou agressivo) correspondente àquele instante de tempo. As subseções a seguir detalham as etapas apresentadas na visão geral.

Figura 4.1: Diagrama da abordagem proposta.



Fonte: Elaborado pelo Autor, 2025.

4.1 Dados dos Sensores

Os dados dos sensores constituem a base para a extração de informações utilizadas na classificação do comportamento do motorista. Para este estudo, foram considerados os PIDs de velocidade, rotações por minuto (RPM), carga do motor e posição do acelerador, que são extraídos da rede *Controller Area Network* (CAN) do veículo. Esses parâmetros foram selecionados por sua capacidade de representar a dinâmica operacional do veículo.

As leituras são atualizadas a uma frequência de 1 Hz, equilibrando a responsividade do sistema com as restrições de processamento. Além disso, os dados passam por um processo de normalização para alinhar as escalas das variáveis, como no caso do RPM, que é dividido por um fator de 100. O restante das variáveis permanecem com os valores lidos originalmente.

Esses dados normalizados servem como entrada para as etapas subsequentes do processamento, permitindo explorar as relações entre os parâmetros monitorados e representando a condição operacional do veículo de forma consolidada.

4.2 Área do Radar

A Etapa 2 da abordagem proposta utiliza os dados normalizados de velocidade, RPM, carga do motor e posição do acelerador para gerar um sensor virtual, o *soft sensor*, que sintetiza as informações de forma gráfica. Cada eixo do gráfico de radar corresponde a uma leitura específica, e a área do polígono formado por essas leituras é calculada utilizando as áreas parciais dos triângulos que compõem o gráfico.

Conforme demonstrado na Figura 4.2, essa área fornece uma métrica dinâmica que reflete o estado operacional instantâneo do veículo, ajustada aos padrões de condução de cada motorista. A métrica é recalculada a cada segundo, acompanhando a frequência de atualização das leituras dos sensores. A saída desta etapa é uma entrada relevante para as etapas posteriores da análise, permitindo capturar a utilização dos recursos do veículo por meio do comportamento do motorista em tempo real.

Figura 4.2: Cálculo da área do Radar.

Fonte: Elaborado pelo Autor, 2025.

O comportamento de direção é refletido nas variações da área total do gráfico. Por exemplo, em uma condução agressiva, caracterizada por acelerações rápidas, observa-se um aumento na carga do motor e na rotação por minuto (RPM), o que resulta em uma área de radar maior. Em contrapartida, uma condução mais estável e suave, como a manutenção de uma velocidade constante de 100 km/h em uma rodovia, tende a apresentar menores níveis de carga do motor e RPM, culminando em uma área reduzida. Destaca-se que variações na área do radar para o mesmo motorista podem ser influenciadas não apenas pelo tráfego ou condições da via, mas também pelas características específicas do modelo do veículo.

Ademais, a interpretação dos dados do gráfico de radar deve considerar fatores relacionados ao veículo, como o tamanho do motor e as relações de marcha, que afetam diretamente as leituras dos sensores. Assim, os resultados obtidos pelo *soft sensor* apresentam maior relevância quando analisados no contexto das particularidades de cada veículo, garantindo que a área do radar represente de forma precisa a utilização de recursos vinculada ao comportamento do motorista.

4.3 MMCloud

O algoritmo MMCloud é um método de clusterização incremental online projetado para agrupar dados em tempo real e que se adapta dinamicamente à estrutura dos dados conforme novos pontos são recebidos. As subseções a seguir detalham, respectivamente, os parâmetros, as variáveis internas do algoritmo e o seu passo-a-passo de execução.

4.3.1 Parâmetros e variáveis internas

A classe do algoritmo MMCloud possui dois hiperparâmetros configuráveis:

1. O número máximo de *clusters* que podem ser criados ao longo da execução, denotado por K_{max};

2. A dimensão dos dados que serão agrupados, denotada por D.

Como deseja-se agrupar os dados com base na área do radar e na carga do motor, foram definidos os valores $D = 2$ e $K_{max} = 3$, uma vez que deseja-se rotular entre cauteloso, normal e agressivo. Além disso, esta classe possui ainda outras variáveis internas que servem para garantir o funcionamento correto do algoritmo. São elas:

1. C: vetor (lista) de *clusters*, onde cada elemento é uma instância da classe *Cluster*, detalhada a seguir. Valor padrão: lista vazia;

2. i: inteiro contendo o índice de quantos *clusters* foram criados até o momento. Valor padrão: 0;

3. n_d: inteiro que armazena a quantidade de distâncias já processadas pelo algoritmo. Cada distância corresponde ao processamento de um ponto $\mathbf{x}$, representando, assim, também, quantos pontos foram efetivamente processados. Valor padrão: 0;

4. μ_d: representa a média (μ) das distâncias euclidianas dos pontos de dados até o centroide do cluster mais próximo, calculada incrementalmente à medida que novos pontos são processados. Esta abordagem evita a necessidade de armazenar todas as distâncias, sendo ideal para cenários com restrições de memória. Valor padrão: 0,0.

5. σ_d^2: representa a variância (σ^2) das distâncias dos pontos de dados até o centroide do cluster mais próximo, também calculada de forma incremental. Similar à média incremental, este cálculo evita o armazenamento de todas as distâncias, otimizando o uso de memória. Valor padrão: 0,0.

Como mencionado anteriormente, o MMCloud possui um vetor *C* que armazena instâncias da classe *Cluster*. Essa classe possui dois parâmetros de inicialização: o *id*, para identificar unicamente cada *cluster*, e a dimensão *D* dos pontos que serão armazenados. O valor de *D* é o mesmo utilizado na inicialização do MMCloud. Além dos parâmetros de inicialização, esta classe também possui algumas variáveis internas. São elas:

1. n_k: inteiro que armazena a quantidade de pontos armazenados naquele *cluster*. Valor padrão: 0;

2. μ_k: vetor de dimensão *D* que representa a média (centro) do *cluster*. É calculada de forma incremental. Valor padrão: vetor de zeros;

3. σ_k^2: vetor de dimensão *D* que representa a variância do *cluster*, também calculada incrementalmente. Valor padrão: vetor de zeros;

4. CV_k: vetor de dimensão *D* que representa o coeficiente de variação do *cluster*, obtido pela divisão entre σ_k^2 e μ_k.Valor padrão: vetor de zeros;

5. l_k: indica o rótulo do *cluster*. Valor padrão: `None`.

Uma vez que todas as variáveis de cada classe foram definidas. Torna-se possível entender com maior clareza o seu funcionamento.

4.3.2 Passo-a-passo de execução

Um único *cluster* é criado na inicialização, servindo como ponto de partida para o processo de clusterização. Para cada novo ponto de dados **x** recebido, o algoritmo realiza os seguintes passos:

1. **Teste do Cluster Mais Próximo**: calcula-se a mínima distância euclidiana (d_{min}) entre o ponto **x** e o centro de cada *cluster k* existente, representado pela média dos pontos armazenados (μ_k), como observa-se na Equação 4.1.

$$d_{\min} = \arg\min_k |\mathbf{x} - \mu_k|. \tag{4.1}$$

Uma vez que $d_{\min}$ é calculado, ela é armazenada para uso posterior.

2. **Atualização das Estatísticas Incrementais de Distância**: as estatísticas de distância são atualizadas sem a necessidade de armazenar todas as distâncias calculadas. A média μ_d e a variância σ_d^2 são atualizadas, respectivamente, usando as Equações 4.2 e 4.3:

$$\mu_{d,\text{new}} = \mu_{d,\text{old}} + \frac{(d_{\min} - \mu_{d,\text{old}})}{n_d + 1}, \tag{4.2}$$

$$\sigma^2_{d,\text{new}} = \sigma^2_{d,\text{old}} + \big(d_{\min} - \mu_{d,\text{old}}\big)\big(d_{\min} - \mu_{d,\text{new}}\big). \tag{4.3}$$

Uma vez que a média e a variância são calculados, o contador *n_distance* é incrementado em 1.

3. **Detecção Dinâmica de *Outliers***: com o intuito de identificar *outliers* de forma adaptativa, implementou-se um limiar dinâmico com base na média μ_d e no desvio-padrão das distâncias ($\sqrt{\sigma_d^2}$). Tal limiar é calculado conforme a Equação 4.4.

$$\delta_{\text{outlier}} = \mu_d + 2,54 \times \sqrt{\frac{\sigma_d^2}{n_d}}. \tag{4.4}$$

Ou seja, se $d_{\min}$ for maior do que $\delta_{outlier}$ e o número atual de *clusters* for menor que K_{max}, cria-se um novo *cluster* para acomodar o ponto $\mathbf{x}$. Caso contrário, se o número máximo de *clusters* já foi atingido, o ponto $\mathbf{x}$ é marcado como *outlier*. Deste modo, conforme abordado na Seção 2.1.3, o fator de 2,54 desvio-padrões representa uma confiabilidade de 98,89% na distribuição dos dados.

4. **Atualização do *Cluster* Atribuído**: os passos anteriores consistiram apenas em testes estatísticos para verificar qual o *cluster* mais próximo e se aquele ponto é ou não um *outlier*. Se o ponto não for considerado *outlier*, ele é adicionado ao *cluster* mais próximo, e as estatísticas internas desse *cluster* são atualizadas conforme as Equações 4.5, 4.6 e 4.7, onde $\mu_{k,\text{old}}$ é a média do *cluster k* antes da atualização.

Contagem de Pontos:

$$n_k \leftarrow n_k + 1 \tag{4.5}$$

Média:

$$\mu_{k,new} \leftarrow \mu_{k,old} + \frac{(\mathbf{x} - \mu_{k,old})}{n_k} \tag{4.6}$$

Variância:

$$\sigma^2_{k,new} \leftarrow \frac{(n_k - 2)\sigma^2_{k,old} + \big(\mathbf{x} - \mu_{k,\text{old}}\big)\big(\mathbf{x} - \mu_{k,new}\big)}{n_k - 1} \tag{4.7}$$

A formulação da Equação 4.7 representa uma abordagem de cálculo incremental para a variância amostral, que é particularmente adequada para ambientes de fluxo de dados. Esta metodologia, frequentemente associada a algoritmos online, permite a atualização contínua da variância sem a necessidade de armazenar o histórico completo dos pontos de dados, o que otimiza o uso de memória e a eficiência computacional. O termo $(n_k - 2)$ e a divisão por $(n_k - 1)$ são elementos inerentes a esta formulação específica para garantir uma estimativa não viciada da variância da amostra (Wei et al. 2023).

5. **Verificação de Dispersão e Divisão de *Clusters***: uma vez que o ponto é adicionado a um *cluster*, verifica-se se os *clusters* estão muito dispersos. Para isso, implementou-se um limiar dinâmico de dispersão seguindo a mesma lógica do limiar de *outlier*, conforme a Equação 4.8:

$$\delta_{\text{disp}} = \mu_d + 2,54 \times \sqrt{\frac{\sigma_d^2}{n_d}}. \tag{4.8}$$

Para avaliar a dispersão de um *cluster*, é calculada a soma de suas variâncias em todas as dimensões, denotada por $\sum_d \sigma_{k,d}^2$. Se $\sum_d \sigma_{k,d}^2$ exceder δ_{disp} e o número atual de *clusters* for menor que K_{max}, o *cluster* é considerado disperso e dividido em dois novos *clusters*. A divisão é realizada ao longo da dimensão d^* com maior variância σ_{k,d^*}^2, onde um dos centros é deslocado em $-0,5\sigma_{k,d^*}^2$ e o outro em $+0,5\sigma_{k,d^*}^2$, conforme as Equações 4.9 e 4.10:

$$\mu_{k_1} = \mu_k, \quad \mu_{k_1,d^*} \leftarrow \mu_{k,d^*} - 0.5 \times \sigma_{k,d^*}^2 \tag{4.9}$$

$$\mu_{k_2} = \mu_k, \quad \mu_{k_2,d^*} \leftarrow \mu_{k,d^*} + 0.5 \times \sigma_{k,d^*}^2 \tag{4.10}$$

O valor de meio desvio-padrão foi definido empiricamente de forma a não tornar os *clusters* tão distantes a ponto de os próximos pontos serem considerados *outliers*, nem tão próximos a ponto de ficarem sobrepostos. Uma vez divididos, o antigo *cluster* é removido de C e os dois novos são adicionados, cada um com seu respectivo *id*.

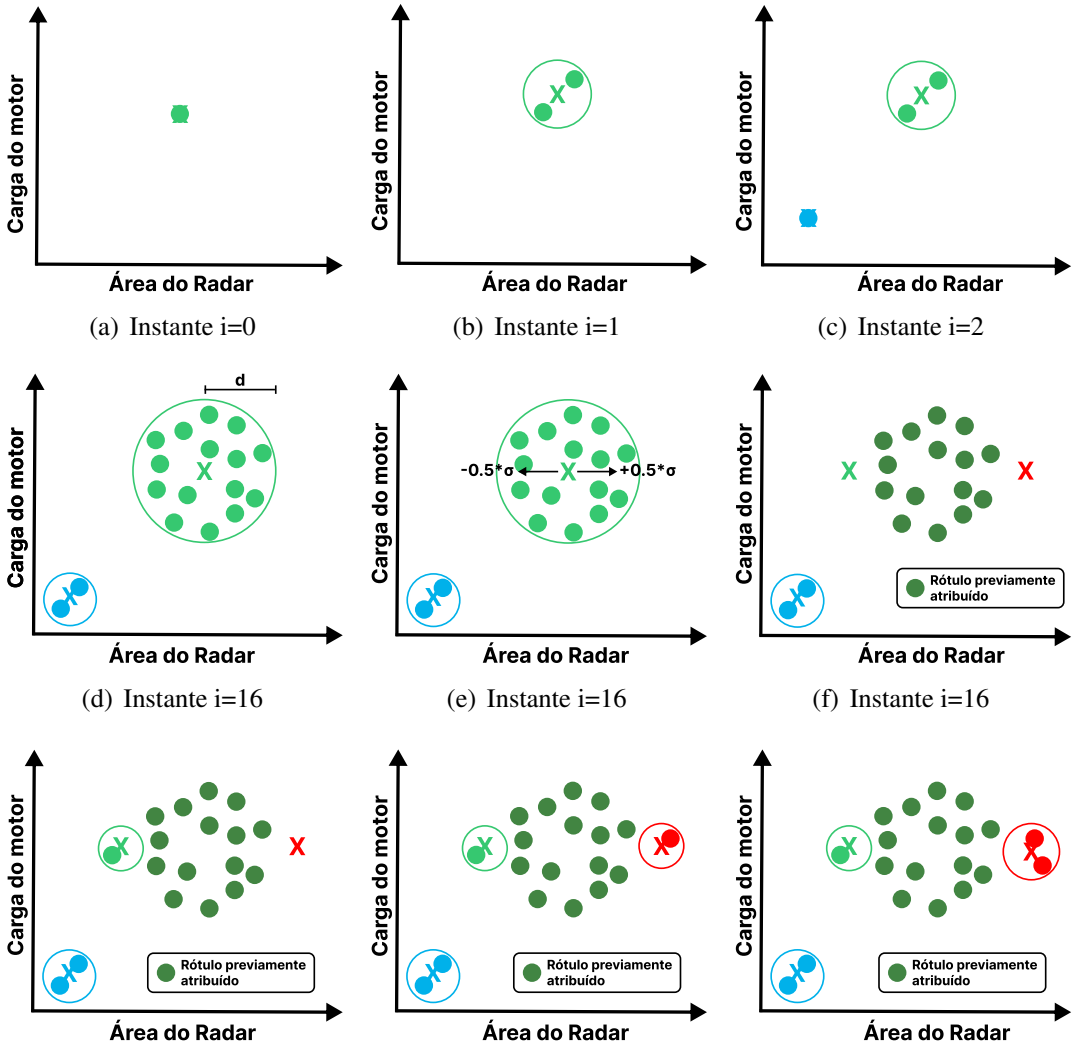
6. **Atribuição dos Rótulos aos *Clusters***: após a atualização das estatísticas dos *clusters* e eventuais divisões, os rótulos são definidos com base na norma dos vetores médios ($|\mu_k|$) de cada *cluster*. Essa etapa organiza os *clusters* existentes em categorias que representam os padrões de comportamento do motorista. A atribuição específica dos rótulos é detalhada na subseção 4.4. A atribuição de rótulos ocorre de maneira contínua e é integrada ao fluxo incremental do MMCloud. Assim, o algoritmo não apenas organiza os dados em *clusters*, mas também vincula esses grupos aos padrões de comportamento predefinidos. A separação lógica desta etapa em uma subseção específica visa descrever a metodologia de forma mais clara e modular.

7. **Conclusão do Ciclo de Atualização**: O ciclo de processamento para cada novo ponto é finalizado após a atribuição de rótulos. O algoritmo mantém a consistência dos *clusters* e garante a atualização incremental das estatísticas internas, permitindo o processamento contínuo de fluxos de dados em tempo real.

A abordagem incremental do MMCloud evita a necessidade de armazenar todo o conjunto de dados ou recalcular estatísticas globais a cada novo ponto, o que é vantajoso em cenários com fluxos de dados contínuos. Além disso, ao limitar o número máximo de *clusters*, o algoritmo mantém a complexidade computacional controlada, equilibrando a expressividade do modelo com a eficiência operacional. Para representar todo o processo, a Figura 4.3 visa demonstrar o processo de execução do algoritmo de forma detalhada.

Assim, na Figura 4.3 (a) no instante inicial ($i = 0$), o primeiro ponto do conjunto de dados é considerado o centro de um novo *cluster*. Esse *cluster* é rotulado como “Normal”

Figura 4.3: Análise dos instantes de simulação.



(a) Instante i=0 (b) Instante i=1 (c) Instante i=2

(d) Instante i=16 (e) Instante i=16 (f) Instante i=16

(g) Instante i=17 (h) Instante i=18 (i) Instante i=19

Fonte: Elaborado pelo Autor, 2025.

e representado pela cor verde. na Figura 4.3 (b) representa o segundo ponto onde ($i = 1$), devido à variância do *cluster* ser igual a zero, ele também é atribuído ao *cluster* "Normal".

No instante $i = 2$ (Figura 4.3 (c)), um novo ponto chega cuja distância em relação ao centro do *cluster* existente excede 2,54 desvios-padrões da média do *cluster*. Por esse motivo, ele forma um novo *cluster*. Como ele é situado mais próximo da origem, ele é rotulado como "Cauteloso" e é representado pela cor azul.

Ao alcançar o instante $i = 16$ (Figura 4.3 (d)), há uma significativa dispersão no *cluster* "Normal", devido ao aumento do número de pontos atribuídos a ele. Se a raiz quadrada da distância (d) exceder 2,54 desvios-padrões da média e ainda houver a possibilidade de criação de novos *clusters*, será realizada a divisão do *cluster*. Após a divisão, os novos centros são deslocados para $-0,5$ desvios-padrões e $+0,5$ desvios-padrões em relação ao

centro original (Figura 4.3 (e)). No exemplo dado, o *cluster* deslocado à direita é rotulado como "Agressivo" e representado pela cor vermelha (Figura 4.3 (f)). É importante destacar que os pontos anteriormente atribuídos aos *clusters* não são redistribuídos após a divisão, mantendo os rótulos previamente atribuídos, ilustrados pelo verde mais escuro na Figura 4.3. Os dois novos *clusters* iniciam sem pontos associados, pois permitem manter a eficiência computacional do algoritmo.

Na Figura 4.3 (g), no instante $i = 17$, um exemplo é ilustrado onde um novo ponto foi adicionado ao *cluster* "Normal", com a variância recalculada em função dessa atualização. No instante $i = 18$, um ponto foi atribuído ao *cluster* "Agressivo", e a variância correspondente também foi recalculada (Figura 4.3 (h)). Por fim, no instante $i = 19$, o novo ponto foi agrupado no *cluster* "Agressivo" (Figura 4.3 (i)), seguindo o processo de clusterização .

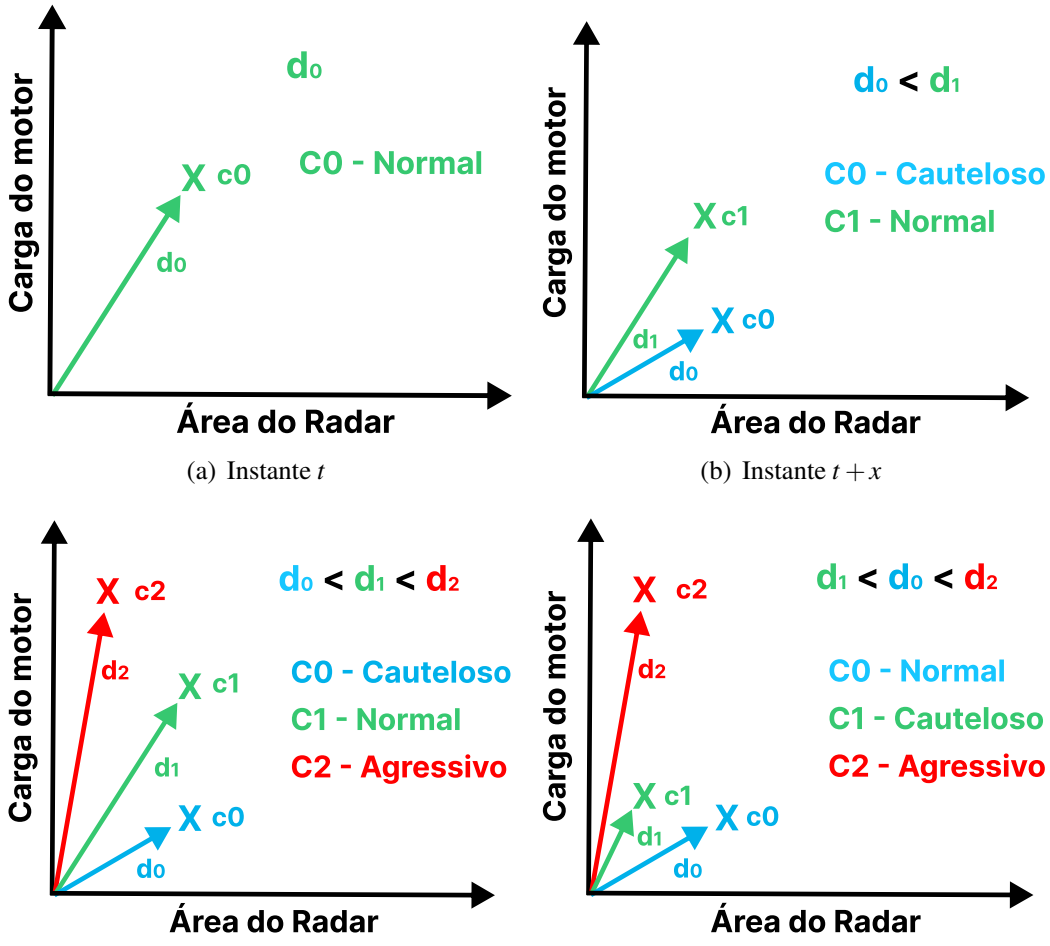
4.4 Rotulação de Dados

A etapa 4, ilustrada na Figura 4.4, realiza a rotulação dos *clusters* gerados pelo MM-Cloud, organizando-os em categorias de comportamento do motorista: "Cauteloso", "Normal" e "Agressivo". A rotulação é fundamentada na posição relativa dos centróides dos *clusters* no espaço vetorial, representados na figura pelo "X", e utiliza a norma das distâncias dos centróides em relação à origem ($|\mu_k|$). Os instantes t, $t+x$, $t+y$ e $t+z$ representam diferentes pontos no tempo do fluxo de dados, demonstrando a adaptação dinâmica do algoritmo a distintas configurações de clusters que podem emergir.

Quando apenas um *cluster* é formado, ele é rotulado como "Normal", conforme indicado na 4.4 (a), onde o *cluster* único ($C0$) é acompanhado por sua norma (d_0). Nos casos em que dois *clusters* são formados, o *cluster* com menor norma é rotulado como "Cauteloso", enquanto o outro é identificado como "Normal", como demonstrado na 4.4 (b), onde as normas (d_0 e d_1) determinam os rótulos $C0$ e $C1$. No cenário com três *clusters*, os rótulos são atribuídos em ordem crescente das normas, sendo o *cluster* com menor norma identificado como "Cauteloso", o intermediário como "Normal" e o maior como "Agressivo". A 4.4 (c) ilustra que, neste caso, as normas (d_0, d_1, d_2) são usadas para distinguir os *clusters* $C0$, $C1$ e $C2$, porém a ordem dos rótulos pode variar de acordo com a disposição dos *clusters* no espaço.

A definição dos rótulos ocorre a partir das relações geométricas entre os *clusters*, sem que haja uma correspondência fixa para cada um deles. Por exemplo, no instante $t+y$ da 4.4 (c), $C0$, $C1$ e $C2$ foram rotulados, respectivamente, como "Cauteloso", "Normal" e "Agressivo". No instante $t+z$, o centróide do *cluster* $C1$ estava mais próximo da origem do que $C0$ e, por isso, foi rotulado como "Cauteloso", enquanto que $C0$ foi rotulado como "Normal" (4.4 (d)). Essa metodologia permite identificar categorias de comportamento de acordo com as distâncias observadas, promovendo consistência na análise de diferentes distribuições de dados.

Figura 4.4: Evolução da formação de clusters ao longo do tempo na etapa 4 da abordagem proposta.



(a) Instante t (b) Instante $t+x$

(c) Instante $t+y$ (d) Instante $t+z$

Fonte: Elaborado pelo Autor, 2025.

Seção 5

Estudo de Caso

Esta seção descreve dois estudos de caso para avaliação da abordagem proposta, com foco na viabilidade de identificar o comportamento do motorista em diferentes cenários de coleta de dados. As próximas subseções detalham a definição e o planejamento de ambos os estudos de caso.

5.1 Planejamento

Esta seção descreve o planejamento detalhado e o *design* dos estudos de caso. Ambos os estudos foram estruturados para garantir uma abordagem sistemática na coleta, análise e interpretação dos dados. Os seguintes aspectos foram considerados na fase de planejamento:

1. **Seleção do contexto**: os estudos de caso terão como foco motoristas em Natal/RN e veículos fabricados no Brasil a partir de 2010, ano em que o sistema OBD-II tornou-se obrigatório nos veículos.

2. **Seleção de variáveis**: para investigar o fenômeno em questão, serão consideradas as seguintes variáveis dependentes, independentes e intervenientes:

    - Variáveis dependentes: sensores do veículo, como RPM, velocidade e posição do acelerador.
    - Variáveis independentes: OBD-II, veículos e rota.
    - Variáveis intervenientes: condições de tráfego, possíveis problemas nos veículos e comportamento *mascarado* dos participantes, ou seja, quando as pessoas simulam um estilo de direção diferente do habitual.

3. **Questões de pesquisa** (QP): as questões a serem exploradas são:

    - **QP1**: como as características individuais do motorista e a familiaridade com o veículo influenciam os padrões de comportamento identificados pelo algoritmo?
    - **QP2**: o tempo médio gasto em cada perfil (cauteloso, normal, agressivo) pode servir como um indicador de estabilidade e transições no comportamento de direção?

- **QP3**: quais são os principais fatores que impactam a qualidade dos *clusters* formados pelo algoritmo MMCloud ao analisar diferentes motoristas e veículos?
- **QP4**: o algoritmo MMCloud é computacionalmente eficiente para aplicações em tempo real em sistemas embarcados?

4. **Seleção de participantes e artefatos**: após definir as questões de pesquisa e variáveis a serem analisadas, iniciou-se o processo de seleção de participantes e objetos para ambos os estudos. Por conveniência, o Departamento de Engenharia de Computação e Automação (DCA/UFRN) foi escolhido para representar a população do estudo.

   Para o **primeiro estudo de caso**, dois motoristas do DCA manifestaram interesse em participar, como mostrado na Tabela 5.1.

   Tabela 5.1: Caracterização dos participantes do primeiro estudo de caso.

Motorista	Marca/Modelo	Ano	Motor
1	Fiat Fastback	2025	1.0L Turbo
2	Volkswagen Polo Comfortline (Branco)	2019	1.0L Turbo

   Para o segundo estudo de caso, utilizando a aplicação APP2Car, um número maior de participantes aceitou integrar o estudo, resultando em uma amostra maior de veículos. A Tabela 5.2 detalha as características desses veículos.

Tabela 5.2: Caracterização dos participantes do segundo estudo de caso (APP2Car).

Motorista	Marca/Modelo	Ano	Motor
1	Mitsubishi Eclipse Cross HPE-S	2025	1.5L Turbo
2	Fiat Fastback	2025	1.0L Turbo
3	Toyota Hilux	2014	3.0L TDI
4	Volkswagen Polo Comfortline (Branco)	2019	1.0L Turbo
5	Volkswagen Polo Comfortline (Prata)	2019	1.0L Turbo
6	Jeep Renegade T270	2024	1.3L Turbo

   Vale destacar que todos os veículos selecionados para este estudo possuíam transmissão automática. Além disso, por razões legais, os nomes dos participantes não serão utilizados neste trabalho. O número do motorista ou a marca/modelo do veículo serão usados para identificar cada indivíduo ou sessão.

5. **Design dos estudos**: ambos os estudos de caso foram conduzidos em Natal, Rio Grande do Norte, Brasil, utilizando uma rota de aproximadamente 6 quilômetros. Esta rota incorpora diversos elementos de tráfego, como rotatórias, semáforos, faixas de pedestres, cruzamentos e curvas com diferentes ângulos (45° e 90°). Essas

características foram selecionadas para garantir que a rota proporcionasse oportunidades de capturar padrões variados de direção, incluindo comportamentos cautelosos, normais e agressivos.

Para o **primeiro estudo de caso**, a coleta de dados ocorreu entre 15h e 16h sob condições moderadas de tráfego para minimizar interferências externas. Foi implementada uma abordagem de validação cruzada, onde cada motorista inicialmente dirigiu seu veículo ao longo da rota definida e, posteriormente, trocou de veículo para repetir a mesma rota. Esse *design* permitiu uma avaliação da influência combinada dos estilos individuais de condução, características do veículo e diferenças nos padrões de direção observados.

Para o **segundo estudo de caso**, a coleta de dados utilizando o aplicativo APP2Car foi realizada em horários comerciais (segunda a sexta) ao longo de uma semana, explorando a mesma rota mencionada anteriormente. Esta abordagem visou capturar uma maior variedade de condições de tráfego e comportamentos.

6. **Instrumentação**: o processo de instrumentação envolveu a configuração de ambientes para ambos os estudos e o planejamento da coleta de dados. A seguir, estão os recursos a serem utilizados:

   a) **Freematics One+**: utilizado no primeiro estudo de caso, é um dispositivo OBD-II projetado para aquisição de dados em tempo real de sensores veiculares (Freematics 2025). Suporta múltiplos protocolos de comunicação e é capaz de coletar uma ampla gama de dados, incluindo leituras de acelerômetro, posição do acelerador, RPM do motor, velocidade do veículo e outras métricas de desempenho. O dispositivo é equipado com capacidades integradas de Bluetooth e registro de dados, permitindo a transmissão contínua dos dados coletados para plataformas externas para análise. Sua versatilidade o torna adequado para aplicações em pesquisa automotiva, análise de comportamento de condução e monitoramento veicular em tempo real.

   b) **Smartphone com Dispositivo OBD-II ELM327**: Utilizado no segundo estudo de caso (APP2Car), esta configuração emprega um smartphone conectado via Bluetooth a um adaptador OBD-II ELM327. O aplicativo APP2Car, desenvolvido especificamente para este estudo, gerencia a comunicação com o ELM327 para coletar dados de sensores do veículo, como RPM, velocidade, posição do acelerador e carga do motor. Vale ressaltar que o mesmo smartphone foi utilizado para registrar as rotas percorridas, garantindo a sincronização dos dados coletados. Essa abordagem permite a coleta de dados em uma plataforma mais acessível e comum, expandindo a aplicabilidade do sistema MMCloud.

   c) **Rota**: Conforme mencionado anteriormente, a rota definida é familiar aos participantes, permitindo que eles dirijam de forma natural e habitual.

   d) **Métricas de Avaliação dos *Clusters***: a avaliação de algoritmos de aprendizado não supervisionado, como os de *clustering*, é desafiadora devido à ausência de rótulos explícitos nos dados. Nesse contexto, métricas baseadas em

critérios internos e externos são amplamente utilizadas (Nordahl et al. 2022, Syahputri et al. 2024, Marrero et al. 2024).

Estas métricas geralmente quantificam dois critérios principais: a **compactação intra-cluster** (quão próximos estão os pontos dentro do mesmo *cluster*) e a **separação inter-cluster** (quão distintos ou distantes estão os diferentes *clusters*).

Para o cálculo de algumas dessas métricas, define-se primeiramente a **Soma dos Quadrados das Distâncias (SQD)** de um *cluster* ω_i em relação a um ponto de referência z (que pode ser um centroide ou outro ponto do *cluster*). Esta medida é dada por:

$$\text{SQD}(z, \omega_i) = \sum_{x_j \in \omega_i} \|x_j - z\|_2^2 \tag{5.1}$$

onde:

- x_j são os pontos de dados pertencentes ao *cluster* ω_i.
- $n_i = |\omega_i|$ é o número de pontos de dados no *cluster* ω_i.
- $\|\cdot\|_2^2$ representa o quadrado da norma Euclidiana (distância Euclidiana ao quadrado). Isso significa que estamos somando as distâncias Euclidianas quadradas de cada ponto x_j no *cluster* ω_i ao ponto z.

Índice Silhouette

Índice Silhouette, proposto por Rousseeuw (1987), quantifica quão similar um objeto é ao seu próprio *cluster* (coesão) em comparação com outros *clusters* (separação). Para um ponto de dado x_k, seu coeficiente de Silhouette $s(x_k)$ é calculado como:

$$s(x_k) = \frac{b(x_k) - a(x_k)}{\max(a(x_k), b(x_k))} \tag{5.2}$$

onde $\omega_{\text{cluster}(k)}$ denota o *cluster* ao qual o ponto x_k pertence:

- $a(x_k)$: É a **distância quadrática média intra-cluster** de x_k para todos os outros pontos dentro do mesmo *cluster* $\omega_{\text{cluster}(k)}$. Este termo representa a coesão do ponto x_k em seu *cluster*. Formalmente, utilizando a definição de SQD:

  $$a(x_k) = \frac{1}{|\omega_{\text{cluster}(k)}| - 1} \text{SQD}(x_k, \omega_{\text{cluster}(k)}) \tag{5.3}$$

  Se o *cluster* $\omega_{\text{cluster}(k)}$ tiver apenas um ponto ($|\omega_{\text{cluster}(k)}| = 1$), $s(x_k)$ é tipicamente definido como 0. Note que $\text{SQD}(x_k, \omega_{\text{cluster}(k)}) = \sum_{x_j \in \omega_{\text{cluster}(k)}} \|x_j - x_k\|_2^2$, e o termo onde $x_j = x_k$ é zero, restando $|\omega_{\text{cluster}(k)}| - 1$ termos não nulos na soma se $|\omega_{\text{cluster}(k)}| > 1$.
- $b(x_k)$: É a **menor distância quadrática média inter-cluster** de x_k para os pontos de qualquer outro *cluster* ω_l (onde $l \neq \text{cluster}(k)$). Este termo

representa a separação do ponto x_k em relação ao *cluster* vizinho mais próximo.

$$b(x_k) = \min_{l \neq \text{cluster}(k)} \left(\frac{1}{|\omega_l|} \text{SQD}(x_k, \omega_l) \right) \tag{5.4}$$

O Índice Silhouette (SIL) global para um conjunto de dados com N amostras é a média dos coeficientes $s(x_k)$ sobre todas as amostras:

$$\text{SIL} = \frac{1}{N} \sum_{k=1}^{N} s(x_k) \tag{5.5}$$

O valor do SIL varia de -1 a 1. Valores próximos a 1 indicam que os pontos estão bem agrupados e os *clusters* estão bem separados. Valores próximos a 0 sugerem *clusters* sobrepostos, e valores negativos indicam que os pontos podem ter sido atribuídos ao *cluster* errado (Rousseeuw 1987, Da Silva et al. 2020).

Índice Davies-Bouldin

O Índice Davies-Bouldin (DB) (Davies & Bouldin, 1979) é uma métrica interna que visa identificar *clusters* que são compactos e bem separados. Ele é calculado como a média da similaridade de cada *cluster* com seu *cluster* mais similar. A similaridade entre dois *clusters* ω_i e ω_j é baseada na razão entre a soma das suas dispersões internas e a distância entre seus centroides. O índice é definido como:

$$\text{DB} = \frac{1}{C} \sum_{i=1}^{C} \max_{j \neq i} \left(\frac{S_i + S_j}{M_{i,j}} \right) \tag{5.6}$$

onde:

- C é o número total de *clusters*.
- S_l: É uma medida da **dispersão intra-cluster** (ou compacidade) do *cluster* ω_l. É a raiz quadrada da distância quadrática média dos pontos do *cluster* ao seu centroide v_l:

$$S_l = \left(\frac{1}{n_l} \sum_{x_k \in \omega_l} \|x_k - v_l\|_2^2 \right)^{\frac{1}{2}} = \sqrt{\frac{\text{SQD}(v_l, \omega_l)}{n_l}} \tag{5.7}$$

  Aqui, v_l é o centroide do *cluster* ω_l, e $n_l = |\omega_l|$.
- $M_{i,j}$: É a **distância inter-cluster** entre os centroides v_i e v_j dos *clusters* ω_i e ω_j, respectivamente, utilizando a norma Euclidiana:

$$M_{i,j} = \|v_i - v_j\|_2 \tag{5.8}$$

Valores menores do índice DB indicam uma melhor partição, ou seja, *clusters* mais compactos e mais distantes entre si (Davies & Bouldin 1979, Da Silva

et al. 2020).

Índice Calinski-Harabasz

O Índice Calinski-Harabasz (CH) (Calinski & Harabasz, 1974), também conhecido como Critério da Razão de Variância (VRC, do inglês, *Variance Ratio Criterion*), avalia a qualidade do agrupamento pela razão entre a dispersão inter-cluster e a dispersão intra-cluster. O índice é definido como:

$$\mathrm{CH} = \frac{\mathrm{BGSS}/(C-1)}{\mathrm{WGSS}/(N-C)} \tag{5.9}$$

onde:

- N é o número total de pontos de dados.
- C é o número total de *clusters*.
- BGSS (Between-Group Sum of Squares): É a soma ponderada dos quadrados das distâncias entre os centroides dos *clusters* e o centroide global dos dados. Mede a separação entre os *clusters*.

$$\mathrm{BGSS} = \sum_{i=1}^{C} n_i \|v_i - \mu_{\mathrm{data}}\|_2^2 \tag{5.10}$$

  onde v_i é o centroide do *cluster* ω_i, $n_i = |\omega_i|$ é o número de pontos em ω_i, e μ_{data} é o centroide de todos os N pontos de dados (centro geométrico dos dados).
- WGSS (Within-Group Sum of Squares): É a soma dos quadrados das distâncias entre os pontos de cada *cluster* e seus respectivos centroides. Mede a compactação dos *clusters*.

$$\mathrm{WGSS} = \sum_{i=1}^{C} \sum_{x_k \in \omega_i} \|x_k - v_i\|_2^2 = \sum_{i=1}^{C} \mathrm{SQD}(v_i, \omega_i) \tag{5.11}$$

Valores maiores do índice CH indicam uma melhor qualidade de agrupamento, com *clusters* mais densos (compactos) e bem separados (Caliński & Harabasz 1974, Da Silva et al. 2020).

Índice de Dunn

O Índice de Dunn (Dunn, 1973) visa identificar *clusters* que são compactos e bem separados. Ele é definido como a razão entre a menor distância inter-cluster e o maior diâmetro intra-cluster. O índice é definido como:

$$\mathrm{DI} = \frac{\min_{1 \le i < j \le C} \delta(\omega_i, \omega_j)}{\max_{1 \le k \le C} \Delta(\omega_k)} \tag{5.12}$$

onde:

- C é o número total de *clusters*.
- $\delta(\omega_i, \omega_j)$: Mede a **separação inter-cluster** entre o *cluster* ω_i e o *cluster* ω_j. Esta é a distância mínima Euclidiana entre quaisquer dois pontos pertencentes a estes dois *clusters* diferentes:

$$\delta(\omega_i, \omega_j) = \min_{x \in \omega_i, y \in \omega_j} \|x - y\|_2 \tag{5.13}$$

- $\Delta(\omega_k)$: Mede o diâmetro intra-cluster do *cluster* ω_k. Este é a maior distância Euclidiana entre quaisquer dois pontos dentro do mesmo *cluster* ω_k:

$$\Delta(\omega_k) = \max_{x,y \in \omega_k} \|x - y\|_2 \tag{5.14}$$

Valores maiores do Índice de Dunn indicam uma melhor qualidade de agrupamento: *clusters* mais compactos (menor diâmetro máximo) e mais bem separados (maior distância mínima entre *clusters*) (Dunn 1973, Da Silva et al. 2020).

5.2 Operação

Esta seção descreve o processo de execução dos estudos de caso, elaborados para garantir procedimentos sistemáticos e consistentes, assegurando, assim, a confiabilidade dos dados coletados ao longo dos estudos. As seguintes fases serão detalhadas para cada estudo:

1. **Preparação**: em ambos os estudos, os voluntários foram recebidos e o objetivo principal do estudo foi apresentado, sem mencionar que se tratava de uma análise do comportamento do motorista. Eles também preencheram o termo de consentimento, que igualmente não indicava que o estudo visava a análise do comportamento do motorista. Para o primeiro estudo de caso, os dispositivos Freematics One+ foram conectados aos veículos. Para o segundo estudo de caso, o aplicativo APP2Car foi configurado em um smartphone Android e conectado via Bluetooth a um dispositivo OBD-II ELM327 nos veículos.

2. **Execução**: Após a realização das etapas anteriores, os estudos de caso foram iniciados conforme o delineamento apresentado.

5.2.1 Coleta de Dados

O processo de coleta de dados foi projetado para garantir a aquisição, armazenamento e disponibilidade dos dados necessários para a análise dos estudos de caso.

No **primeiro estudo de caso**, antes de cada voluntário iniciar o percurso definido, um dispositivo Freematics One+ foi conectado ao veículo para coletar dados a cada 1

segundo. O dispositivo registra dados de sensores do veículo, incluindo leituras de acelerômetro, posição do acelerador, RPM, velocidade, carga do motor, avanço de tempo do motor, pressão absoluta do coletor, relação ar-combustível, temperatura de admissão, voltagem da bateria e nível de combustível. Durante a viagem, os dados coletados são transmitidos via redes 3G ou 4G e, ao final de cada viagem, os dados são automaticamente enviados para a plataforma Conect2AI (Conect2AI 2025) para armazenamento. Além disso, o algoritmo MMCloud foi incorporado ao dispositivo Freematics One+. Essa implementação embarcada permite o processamento em tempo real dos dados coletados, possibilitando a classificação imediata do comportamento de direção e a identificação de anomalias no pavimento à medida que o veículo percorre a rota definida.

No **segundo estudo de caso**, utilizando o aplicativo APP2Car em um smartphone conectado via Bluetooth a um dispositivo OBD-II ELM327, os dados dos sensores veiculares (RPM, velocidade, posição do acelerador, carga do motor, entre outros) foram coletados e processados. O algoritmo MMCloud foi integrado diretamente ao aplicativo, permitindo que a inferência do comportamento do motorista fosse realizada em tempo real no próprio smartphone, otimizando o fluxo de dados e reduzindo a dependência de conectividade de rede constante. Ao final de cada viagem, os dados processados e os resultados da classificação também foram enviados para a plataforma Conect2AI para armazenamento e posterior análise.

Por fim, para promover a transparência e facilitar futuras colaborações, todo o código desenvolvido para a implementação, incluindo o algoritmo MMCloud, foi disponibilizado publicamente em um repositório no GitHub, acessível em `https://github.com/conect2ai/MMCloud`.

Seção 6

Resultados

Nesta seção, apresentam-se os resultados obtidos a partir do estudo de caso, cujo objetivo foi analisar a abordagem proposta para a análise do comportamento de direção dos motoristas em diferentes veículos. As questões de pesquisa foram abordadas com base nos dados coletados, investigando como as características individuais dos motoristas, a familiaridade com o veículo, os padrões de comportamento e a eficiência computacional do algoritmo influenciam os resultados.

6.1 Resultados no Freematics One+

A abordagem proposta buscou compreender como as características individuais dos motoristas e a familiaridade com o veículo influenciam os padrões de comportamento. Os dados coletados para os motoristas 1 e 2, ao conduzirem os Carros 1 e 2, indicam uma variação no comportamento de direção.

Deste modo, para o Motorista 1 no Carro 1, foi observado um padrão de transições mais rápidas entre os *clusters* “Cauteloso”, “Normal” e “Agressivo”, evidenciando maior flexibilidade na condução (Figura 6.1 (a)). Deste modo, representados no mapa, respectivamente, pelos pontos azuis e vermelhos. Ou seja, o motorista acelera bruscamente o veículo em trechos curtos e, predominantemente, após lombadas e faixas de pedestres. Depois, o motorista retorna para um perfil “Normal”, representado no mapa pelos pontos em verde e logo depois para um perfil “Cauteloso”. Já no Motorista 1 no Carro 2, o comportamento foi mais constante, com maior permanência no *cluster* “Cauteloso”, sugerindo um estilo de direção mais controlado e focado na segurança (Figura 6.1 (b)). Para esse condutor, percebe-se que, apenas em trechos específicos onde o motorista se sentiu confortável para demandar mais do veículo, é que se obteve uma classificação “Agressiva” para o seu perfil.

O Motorista 2 no Carro 1 apresentou uma maior permanência no *cluster* “Cauteloso”, o que indica um comportamento mais conservador (Figura 6.1 (c)). Por outro lado, no Carro 2, o comportamento do Motorista 2 foi mais agressivo, com transições mais frequentes entre o *cluster* “Agressivo” e “Normal” (Figura 6.1 (d)). Nesse sentido, percebe-se que a abordagem foi capaz de identificar e rotular o comportamento do motorista em todos as rodadas do estudo de caso.

Vale destacar que na Figura 6.1 é perceptível alguns pontos de dados ausentes ao longo

Figura 6.1: Análise de rota dos motoristas nos carros.

(a) Motorista 1 no carro 1

(b) Motorista 1 no carro 2

(c) Motorista 2 no carro 1

(d) Motorista 2 no carro 2

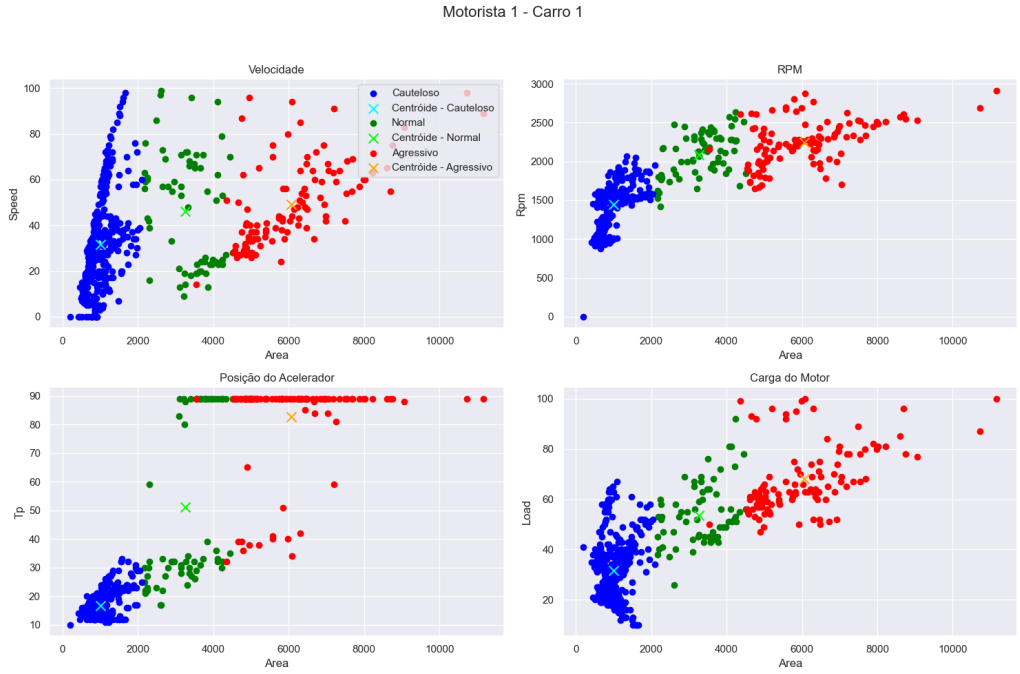
Fonte: Elaborado pelo Autor, 2025.

das rotas devido a lapsos ocasionais na capacidade do *hardware* de registrar informações de latitude e longitude. Apesar dessas lacunas, um exame completo do conjunto de dados revelou que o algoritmo classificou consistentemente o comportamento do motorista durante todo o processo de coleta de dados.

Além disso, foram analisados gráficos de dispersão para examinar o agrupamento dos dados do algoritmo após as viagens, com o eixo Y representando as variáveis do gráfico de radar e o eixo X indicando a área do gráfico de radar. As Figuras 6.2, 6.3, 6.4 e 6.5 ilustram o comportamento do algoritmo em diferentes condições de condução, permitindo analisar as diferenças nos padrões de direção entre motoristas e veículos.

No caso do Motorista 1 no Carro 1 (Figura 6.2), observa-se uma separação entre os *clusters*, com uma pequena sobreposição entre as categorias “Normal” e “Agressivo”. Os centróides desses *clusters* estão bem distribuídos ao longo do eixo da variável “Área”, indicando uma possível correlação com os padrões de condução. O *cluster* “Cauteloso”,

Figura 6.2: *Cluster* final do algoritmo para o Motorista 1 no carro 1.



Fonte: Elaborado pelo Autor, 2025.

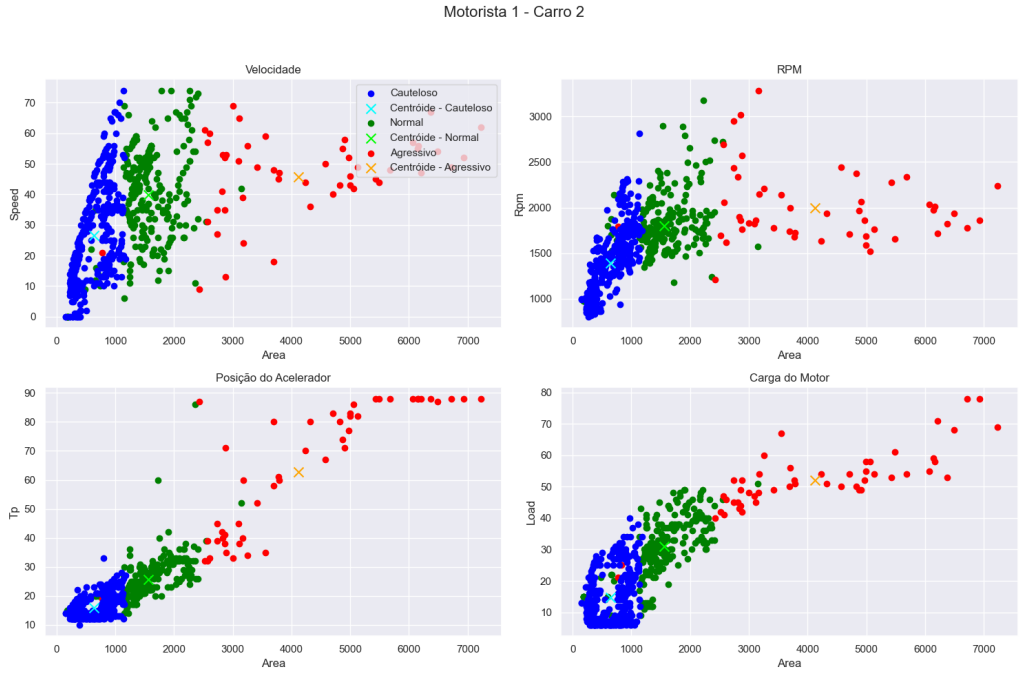
caracterizado por valores baixos de "Área", reflete um estilo mais conservador, enquanto "Normal" representa um padrão intermediário e "Agressivo" está associado a altas rotações e acelerações mais fortes. A sobreposição entre os *clusters* "Normal" e "Agressivo" pode ser atribuída a transições de comportamento, como acelerações rápidas, que são mais evidentes nos gráficos de "Posição do Acelerador" e "Carga do Motor".

No Motorista 1 no Carro 2 (Figura 6.3), os padrões gerais de condução permanecem semelhantes, mas há diferenças na densidade e dispersão dos *clusters*. O *cluster* "Cauteloso" se mostra mais compacto, sugerindo uma maior consistência no estilo conservador do motorista. Em contrapartida, o *cluster* "Normal" exibe maior dispersão, o que pode ser explicado por diferenças na familiaridade do motorista com o veículo ou suas características técnicas. O *cluster* "Agressivo", embora com os maiores valores de "Área" e dispersão, apresenta uma frequência menor, indicando que esse padrão de condução ocorre com menos regularidade.

Por outro lado, no caso do Motorista 2 (Figura 6.4), os resultados indicam padrões de condução distintos. No Carro 1, a predominância do *cluster* "Cauteloso" sugere um estilo mais prudente e conservador, com baixa dispersão e valores menores de "Área". O *cluster* "Normal" apresenta maior dispersão e sobreposição com o *cluster* "Cauteloso", sugerindo transições graduais, especialmente em situações de aceleração. Já o *cluster* "Agressivo" aparece com menor frequência, indicando que o motorista adotou esse padrão de condução de forma isolada e em momentos de maior intensidade.

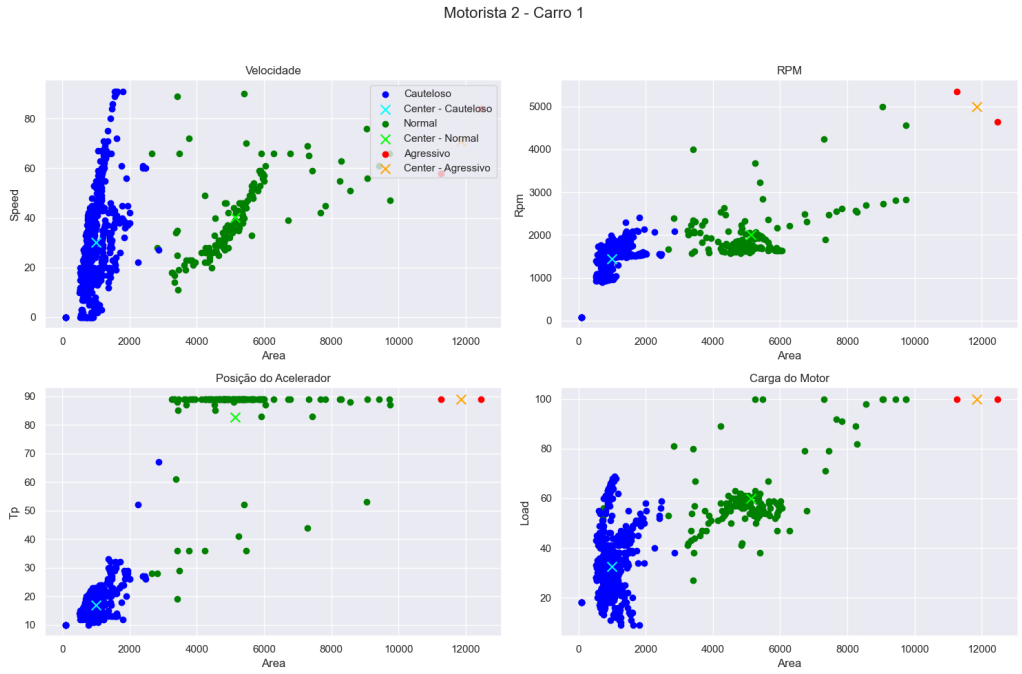
No Motorista 2 no Carro 2, o padrão "Cauteloso" também predomina, mas com uma separação bem definida em relação aos outros *clusters*, o que reforça o estilo conserva-

Figura 6.3: *Cluster* final do algoritmo para o Motorista 1 no carro 2.



Fonte: Elaborado pelo Autor, 2025.

Figura 6.4: *Cluster* final do algoritmo para o Motorista 2 no carro 1.

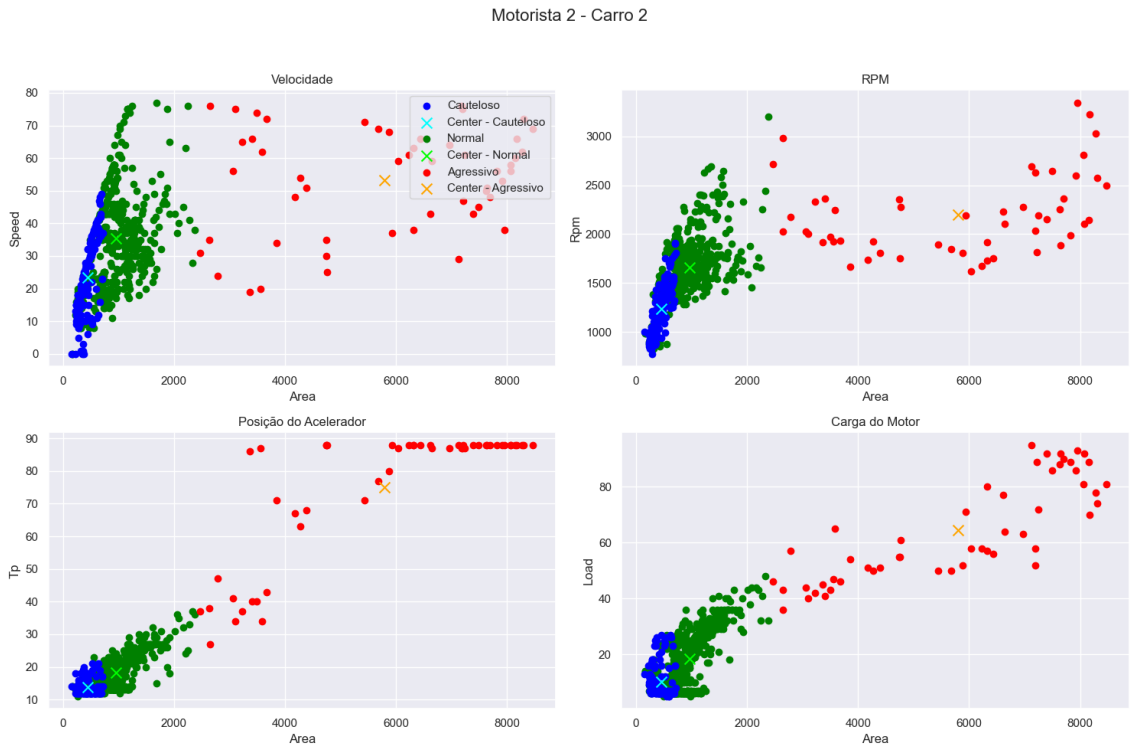


Fonte: Elaborado pelo Autor, 2025.

dor de direção. O *cluster* "Normal" cobre uma faixa mais ampla de valores de "Área",

com sobreposição com o *cluster* “Agressivo”, que, por sua vez, apresenta menor densidade de pontos. Isso sugere uma menor frequência de eventos agressivos, reforçando as variações nos perfis de condução, mesmo quando o motorista está utilizando o mesmo veículo. Esses resultados demonstram que, apesar de utilizarem os mesmos veículos, os motoristas apresentaram padrões de condução distintos, o que evidencia a influência das características individuais no comportamento de direção.

Figura 6.5: *Cluster* final do algoritmo para o Motorista 2 no carro 2.



Fonte: Elaborado pelo Autor, 2025.

A relação entre o tempo médio gasto em cada perfil de direção (“Cauteloso”, “Normal”, “Agressivo”) e a estabilidade do comportamento de direção. A análise dos tempos de permanência nos *clusters*, conforme apresentado na Tabela 6.1, revela que o tempo médio gasto em cada perfil pode, de fato, servir como um indicador importante para avaliar tanto a estabilidade quanto as transições no comportamento de condução.

Para o Motorista 1 - Carro 1, observa-se que o maior tempo foi passado no *cluster* “Cauteloso”, com 423 segundos e um tempo médio de permanência de 16,92 segundos, indicando um comportamento mais estável e consistente. Esse tempo elevado no perfil conservador sugere que, ao adotar um estilo “Cauteloso”, o motorista tende a mantê-lo por períodos mais longos, resultando em menor frequência de transições para outros comportamentos. Por outro lado, o tempo médio de 1,60 segundos no *cluster* “Normal” e 4,04 segundos no *cluster* “Agressivo” indicam que, ao entrar nesses perfis, o motorista muda rapidamente, refletindo uma maior fluidez ou adaptabilidade no estilo de condução.

Tabela 6.1: Métricas de Tempo em Cada *Cluster* para Cada Motorista e Veículo.

Métrica	**Motorista 1 - Carro 1**	**Motorista 1 - Carro 2**	**Motorista 2 - Carro 1**	**Motorista 2 - Carro 2**
Tempo Total em Cada *Cluster* (em segundos)				
Cauteloso (0)	423	362	541	130
Normal (1)	69	232	131	438
Agressivo (2)	109	51	2	52
Tempo Médio por *Cluster* (em segundos)				
Cauteloso (0)	16,92	16,45	28,47	7,22
Normal (1)	1,60	8,29	6,24	18,25
Agressivo (2)	4,04	7,29	1,00	8,67
Tempo Médio Geral Antes de Trocar de *Cluster* (em segundos)				
Tempo Médio	6,33	11,32	16,05	12,92

Fonte: Elaborado pelo Autor, 2025.

O tempo médio geral de 6,33 segundos antes de mudanças de *cluster* reforça a ideia de que o Motorista 1, no Carro 1, tende a alternar com maior frequência entre os diferentes estilos de direção.

No Motorista 1 - Carro 2, o padrão de comportamento é mais equilibrado, com o *cluster* "Cauteloso" ainda predominando, mas com uma maior distribuição de tempo entre os *clusters*. O tempo total de 232 segundos no *cluster* "Normal" e a média de 8,29 segundos indicam que, ao adotar um estilo intermediário, o motorista consegue mantê-lo de forma mais estável, ao contrário do primeiro cenário, onde o tempo de permanência no perfil "Normal" era bem menor. O *cluster* "Agressivo", por sua vez, tem um tempo médio de 7,29 segundos, sugerindo que, quando o motorista assume esse perfil, ele o mantém por um período mais prolongado em comparação ao Motorista 1 no Carro 1, o que implica em transições menos frequentes para outros perfis a partir do "Agressivo". O tempo médio de 11,32 segundos antes das mudanças de *cluster* sugere que a condução no Carro 2 é mais constante e previsível, com menor oscilação entre os diferentes comportamentos.

Para o Motorista 2 - Carro 1, a predominância no *cluster* "Cauteloso", com 541 segundos e uma média de 28,47 segundos. Isso indica um estilo de direção conservador e estável, com permanências prolongadas nesse perfil e, consequentemente, poucas variações nos padrões de comportamento. O tempo muito reduzido no *cluster* "Agressivo" (apenas 2 segundos e uma média de 1 segundo) reforça a ideia de que este motorista raramente adota um estilo mais intenso de condução e, quando o faz, é por um período muito breve. O tempo médio geral de 16,05 segundos antes de mudanças entre os *clusters* é o mais alto entre os casos analisados, sugerindo uma estabilidade ainda maior no comportamento de direção deste motorista, com menor propensão a transições rápidas de perfil.

No Motorista 2 - Carro 2, observa-se uma distribuição mais equilibrada entre os *clusters* "Cauteloso" e "Normal". O *cluster* "Normal" domina com 438 segundos e uma média de 18,25 segundos, o que indica que este motorista mantém um comportamento intermediário de direção de forma estável, com permanências mais longas, mas com mais variação do que o Motorista 2 no Carro 1. O *cluster* "Agressivo" apresenta 52 segundos e uma média de 8,67 segundos, sugerindo que, quando o motorista adota esse perfil, ele o

mantém por um tempo mais prolongado, resultando em transições menos frequentes para outros comportamentos. O tempo médio de 12,92 segundos antes das mudanças entre os *clusters* reforça que o Motorista 2 no Carro 2 tem uma condução relativamente estável, mas ainda mais propensa a oscilações do que no Carro 1.

Portanto, os dados sugerem que quanto maior o tempo médio passado no *cluster* "Cauteloso", maior a estabilidade do comportamento de direção, como evidenciado pelo Motorista 2 no Carro 1 e pelo Motorista 1 no Carro 2. Por outro lado, os motoristas que apresentam maior distribuição de tempo entre os *clusters*, com transições mais rápidas e frequentes, como o Motorista 1 no Carro 1, tendem a ter comportamentos mais instáveis ou adaptáveis, alternando entre os diferentes perfis de direção com maior frequência. Esses resultados indicam que o tempo médio em cada *cluster* não só reflete os estilos de direção dos motoristas, mas também pode ser usado como um indicador de estabilidade e transições no comportamento de direção.

Para entender os principais fatores que impactam a qualidade dos *clusters* formados pelo algoritmo MMCloud ao analisar diferentes motoristas e veículos, avaliamos as métricas de agrupamento, com foco principal nas combinações de "Área" e "Carga do Motor" é apresentada na Tabela 6.2, e revela os principais fatores que influenciam essa qualidade.

Tabela 6.2: Métricas de Clusterização para Cada Motorista e Veículo.

Métrica	**Motorista 1 - Carro 1**	**Motorista 1 - Carro 2**	**Motorista 2 - Carro 1**	**Motorista 2 - Carro 2**
Resultados para Área e Carga do Motor				
Índice de Silhueta	0,7239	0,5288	0,8381	0,2589
Índice de Davies-Bouldin	0,5209	0,6360	0,2799	0,6823
Índice de Calinski-Harabasz	2533,4740	962,3337	1945,7452	1417,4516
Índice de Dunn	0,0006	0,0003	0,0005	0,0000
Resultados para Área e Velocidade				
Índice de Silhueta	0,7236	0,5280	0,8380	0,2582
Índice de Davies-Bouldin	0,5217	0,6366	0,2799	0,6837
Índice de Calinski-Harabasz	2532,0056	961,6932	1945,1208	1417,0264
Índice de Dunn	0,0001	0,0003	0,0002	0,0000
Resultados para Área e RPM				
Índice de Silhueta	0,6833	0,4080	0,8066	0,2626
Índice de Davies-Bouldin	0,5679	0,8044	0,3185	0,8235
Índice de Calinski-Harabasz	2263,7708	709,2032	1587,0529	1167,3814
Índice de Dunn	0,0025	0,0023	0,0007	0,0002
Resultados para Área e Posição do Acelerador				
Índice de Silhueta	0,7242	0,5290	0,8383	0,2592
Índice de Davies-Bouldin	0,5216	0,6359	0,2797	0,6814
Índice de Calinski-Harabasz	2533,7760	962,3788	1946,4522	1417,6872
Índice de Dunn	0,0000	0,0000	0,0001	0,0000

Para o Motorista 1 - Carro 1, os resultados destacam-se pela elevada qualidade do agrupamento. O Silhouette Score de 0,7239 indica uma boa separação entre os *clusters*, sugerindo que as amostras estão bem atribuídas aos seus respectivos grupos. O Davies-Bouldin Score de 0,5209, relativamente baixo, reflete uma baixa similaridade intra-cluster e uma boa separação entre *clusters*. O Calinski-Harabasz Score de 2533,474 reforça esses resultados, indicando que a dispersão entre os *clusters* é substancialmente maior que a dispersão intra-cluster. No entanto, o Dunn Index de 0,0006 aponta que, apesar da boa

separação global, a menor distância relativa entre *clusters* ainda é pequena, possivelmente devido a pontos próximos aos limites entre os grupos.

No caso do Motorista 1 - Carro 2, observa-se uma redução na qualidade dos agrupamentos. O Silhouette Score de 0,5288 sugere maior sobreposição entre os *clusters*, enquanto o Davies-Bouldin Score de 0,636 indica uma menor separação entre os grupos em comparação ao primeiro carro. O Calinski-Harabasz Score de 962,3337, significativamente inferior ao do primeiro veículo, evidencia uma redução na estrutura dos *clusters*, possivelmente influenciada por características específicas do veículo ou diferenças no comportamento do motorista ao conduzi-lo. O Dunn Index, com valor de 0,0003, reforça a proximidade entre *clusters*, o que dificulta a discriminação clara entre eles.

Para o Motorista 2 - Carro 1, as métricas de qualidade de *cluster* indicam uma boa separação e definição dos grupos formados. O Silhouette Score de 0,8381 destaca uma perceptível distinção entre os *clusters*, com as amostras bem alocadas em seus respectivos grupos. O Davies-Bouldin Score de 0,2799, significativamente baixo, reflete uma separação significativa entre os *clusters* e uma baixa similaridade intra-cluster. O Calinski-Harabasz Score de 1945,7452 demonstra uma dispersão substancial entre os *clusters*, reforçando a estrutura bem definida dos agrupamentos. Apesar disso, o Dunn Index de 0,0005 evidencia que a menor distância relativa entre *clusters* ainda é reduzida, sugerindo a presença de alguns pontos próximos às fronteiras dos grupos. Esses resultados ressaltam a eficácia do algoritmo em capturar os padrões de comportamento do Motorista 2 ao dirigir o Carro 1, com forte distinção entre os diferentes perfis de direção.

Já para o Motorista 2 - Carro 2, as métricas apontam para uma qualidade ainda menor no agrupamento. O Silhouette Score de 0,2589 reflete uma fraca separação entre os *clusters*, enquanto o Davies-Bouldin Score de 0,6823 sugere que a similaridade intra-cluster é elevada, dificultando a distinção entre os grupos. O Calinski-Harabasz Score de 1417,4516 apresenta um valor intermediário, mas ainda significativamente inferior ao observado para o Motorista 1 - Carro 1, indicando uma estrutura menos definida dos *clusters*. O Dunn Index de 0,0 evidencia que a menor distância entre os *clusters* não é significativa, reforçando a dificuldade de discriminação em relação a este motorista.

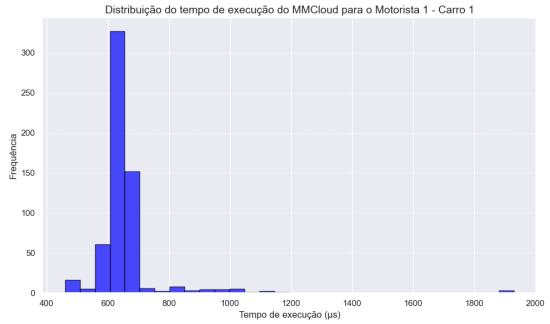
Analisando as outras combinações de variáveis como "Área" e "Velocidade", "Área" e "RPM", bem como "Área" e "Posição do Acelerador", observa-se que, em alguns casos, as métricas apresentam valores mais elevados, sugerindo melhores separações e estruturas em determinados contextos. Por exemplo, os experimentos envolvendo "Área" e "Posição do Acelerador" no Motorista 1 - Carro 1 registram um Silhouette Score de 0,7242, levemente superior à combinação de "Área" e "Carga do Motor", indicando que essa variável pode fornecer informações complementares na análise. No entanto, o Dunn Index permanece consistentemente baixo ou nulo, o que ressalta a dificuldade em estabelecer separações absolutas entre *clusters* em todas as variáveis.

Esses resultados confirmam que, embora "Área" e "Carga do Motor" sejam adequadas para discriminar os *clusters* no contexto geral, fatores como o veículo e o estilo de condução influenciam significativamente a qualidade do agrupamento. As métricas complementares das outras variáveis mostram que é possível explorar diferentes combinações para obter *insights* adicionais sobre o comportamento dos motoristas, mas a discriminação absoluta dos *clusters* ainda enfrenta limitações, especialmente em contextos com maior

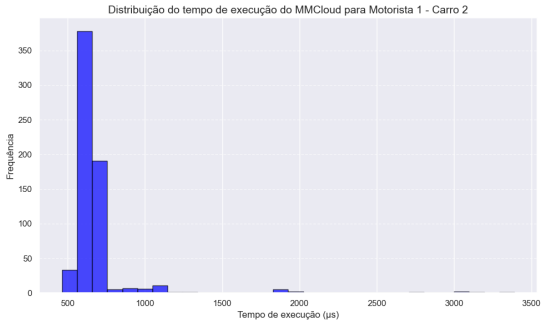
sobreposição entre grupos.

Para avaliar a eficiência computacional do algoritmo MMCloud em sistemas embarcados, foram mensurados os tempos de execução **diretamente no hardware durante os experimentos**. As medições correspondem à diferença de tempo na execução da função principal do algoritmo, responsável por agrupar e retornar o rótulo de cada ponto de dados. Os resultados estão ilustrados nas Figuras 6.6 (a), 6.6 (b), 6.6 (c) e 6.6 (d). Essas figuras mostram a distribuição dos tempos de execução em microsegundos (μs) para cada combinação de motorista e veículo, fornecendo uma visão clara do desempenho do algoritmo.

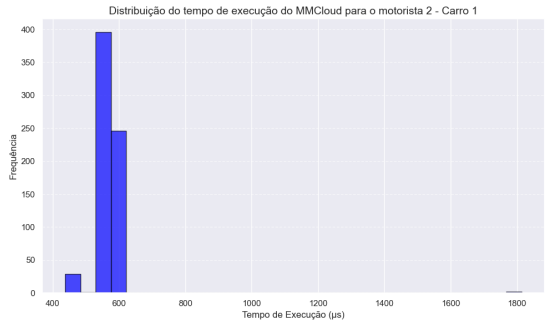
Figura 6.6: Tempos de execução do algoritmo para os motoristas e carros.



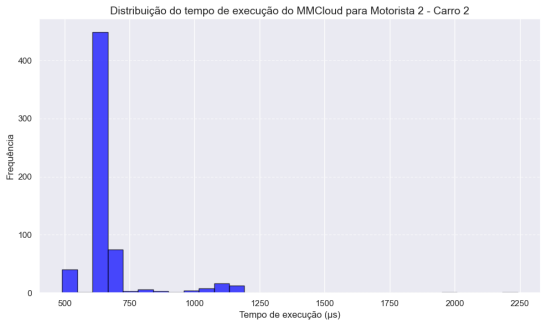
(a) Motorista 1 no Carro 1.



(b) Motorista 1 no Carro 2.



(c) Motorista 2 no Carro 1.



(d) Motorista 2 no Carro 2.

Fonte: Elaborado pelo Autor, 2025.

A partir dos resultados, observa-se que, para o Motorista 1 - Carro 1 (Figura 6.6 (a)), a maioria dos tempos de execução está concentrada entre 600 e 700 μs, com exceções raras que ultrapassam 1000 μs. Esse padrão sugere que o algoritmo apresenta um bom desempenho, com um tempo de execução consistente, o que é desejável para aplicações em tempo real. No entanto, os picos isolados acima de 1500 μs indicam que fatores externos, como a carga do sistema ou variações na dinâmica dos dados, podem ocasionalmente afetar o tempo de processamento.

No caso do Motorista 1 - Carro 2 (Figura 6.6 (b)), a distribuição dos tempos de execução é semelhante, mas com maior variabilidade, atingindo até 3000 μs. Essa dispersão pode ser explicada por características específicas do motorista ou do veículo, que influenciam a complexidade computacional em determinadas condições. Embora essa variação

seja notável, ainda assim, a maior parte das execuções ocorre em tempos aceitáveis para sistemas embarcados.

Para os cenários envolvendo o Motorista 2, como ilustrado nas Figuras 6.6 (c) e 6.6 (d), os tempos de execução mostram uma maior consistência, com a maioria das execuções entre 500 e 700 μs. Isso demonstra que o algoritmo se adapta bem a diferentes perfis de motorista e veículos, mantendo um desempenho eficiente e estável.

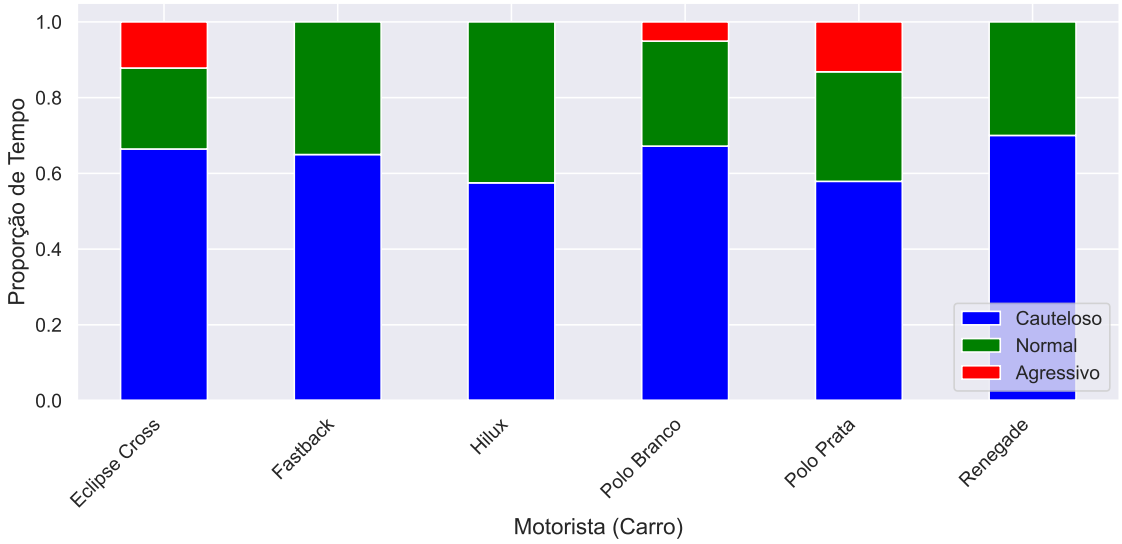
De forma geral, os resultados indicam que o algoritmo MMCloud é computacionalmente eficiente para a maioria das situações analisadas. Embora alguns picos de tempo elevados tenham sido observados, estes são isolados e não comprometem significativamente a viabilidade do algoritmo para aplicações em tempo real em sistemas embarcados.

6.2 Resultados no APP2Car

Esta seção apresenta os resultados obtidos a partir do estudo de caso realizado com o código do MMCloud integrado ao aplicativo APP2Car.

A avaliação da distribuição dos estilos de condução destaca padrões distintos para cada motorista, conforme ilustrado na Figura 6.7. De forma geral, o modo "Normal" predomina na maioria dos condutores, variando entre aproximadamente 40% e 80% do tempo total de condução, sugerindo que grande parte dos motoristas adota um ritmo estável, equilibrando eficiência e segurança.

Figura 6.7: Proporção dos estilos de condução por motorista (APP2Car).



Fonte: Elaborado pelo Autor, 2025.

O estilo "Agressivo" constitui uma minoria, representando menos de 10% do tempo em todas as unidades analisadas. Este baixo percentual pode indicar uma menor incidência de acelerações bruscas e frenagens intensas, refletindo uma conscientização em relação ao consumo de combustível e à segurança veicular.

Em contrapartida, o estilo “Cauteloso” apresenta uma participação intermediária, com variações entre os veículos. Na Hilux e no Polo Prata, por exemplo, o comportamento “Cauteloso” ultrapassa 40% do tempo de condução. Já no Fastback, esse valor se aproxima de 15%. Este contraste pode estar associado ao perfil de cada veículo e ao contexto de uso. Veículos como a Hilux e o Polo Prata, que são uma pick-up e um hatch compacto, respectivamente, podem incentivar uma condução mais conservadora, especialmente em ambientes urbanos com tráfego intenso. Em contraste, SUVs de porte maior, como o Fastback, podem ser conduzidos de forma mais dinâmica, com uma maior proporção do estilo “Normal”.

Analisando individualmente, observa-se que o motorista do Eclipse Cross apresenta uma distribuição equilibrada, com o estilo “Cauteloso” representando cerca de 65% do tempo, o “Normal” aproximadamente 20%, e o “Agressivo” em torno de 15%. O Fastback, por sua vez, exibe uma predominância do estilo “Normal”, correspondendo a cerca de 65% do tempo, enquanto o “Cauteloso” e o “Agressivo” compõem aproximadamente 15% e 5%, respectivamente. Para a Hilux, o estilo “Normal” é mais proeminente, com cerca de 58% do tempo, seguido de perto pelo “Cauteloso”, com aproximadamente 40%, e uma parcela mínima do “Agressivo”, em torno de 2%.

O Polo Branco demonstra uma inclinação maior para o estilo “Normal”, abrangendo aproximadamente 68% do tempo, com o “Cauteloso” em torno de 28% e o “Agressivo” em cerca de 4%. Similarmente, o Polo Prata exibe uma distribuição em que o estilo “Cauteloso” e “Normal” são os mais representativos, com cerca de 58% para o “Cauteloso” e 30% para o “Normal”, e o “Agressivo” contribuindo com aproximadamente 12%. Por fim, o Renegade apresenta uma predominância do estilo “Normal”, com aproximadamente 70% do tempo, seguido pelo “Cauteloso” com cerca de 28%, e o “Agressivo” com uma proporção mínima de 2%.

Em síntese, os resultados apontam para um domínio consistente do estilo “Normal” na maioria dos motoristas e veículos, uma baixa ocorrência de práticas agressivas e uma variação no estilo “Cauteloso” que parece ser influenciada pelas características do veículo, sugerindo que tanto fatores comportamentais dos motoristas quanto atributos do próprio automóvel contribuem para as escolhas de condução.

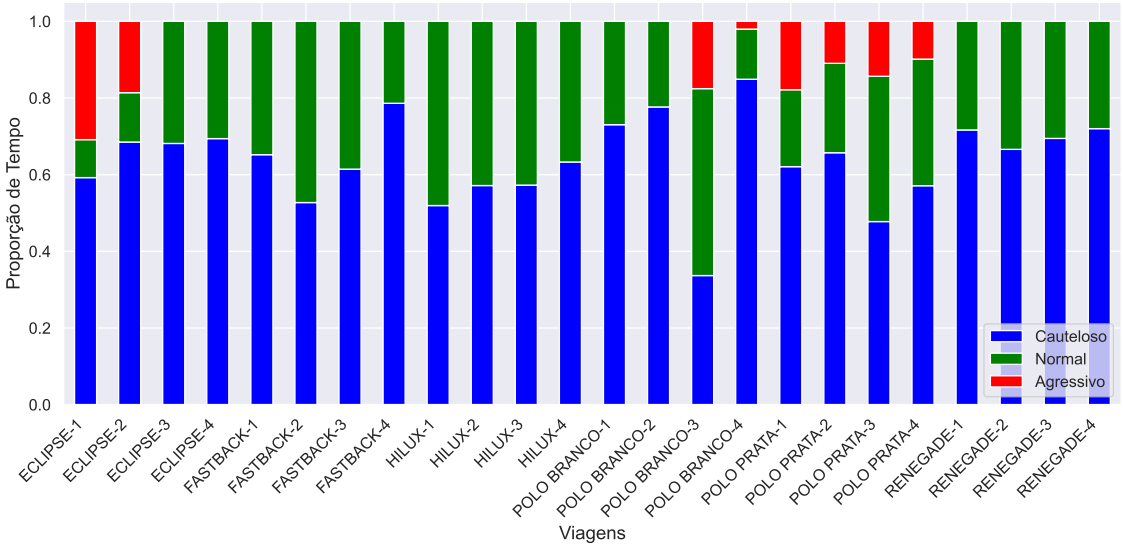
6.2.1 Análise por Viagem

Dando continuidade à análise dos estilos de condução, a Figura 6.8 apresenta a distribuição proporcional dos estilos de condução para cada viagem realizada com os veículos Eclipse Cross, Fastback, Hilux, Polo Branco, Polo Prata e Renegade. Esta análise individualizada por viagem permite observar a variabilidade e consistência dos padrões de condução em diferentes contextos operacionais.

Para o Eclipse Cross, as viagens ECLIPSE-1 e ECLIPSE-2, realizadas no mesmo dia, mostram a presença dos três estilos de condução, com uma proporção notável de estilo “Agressivo”. Nas viagens subsequentes, ECLIPSE-3 e ECLIPSE-4, o estilo “Agressivo” não é identificado, e o estilo “Cauteloso” torna-se predominante, sugerindo uma mudança no perfil de condução ou adaptação às condições da via.

No Fastback, as quatro viagens (FASTBACK-1 a FASTBACK-4) apresentam predo-

Figura 6.8: Proporção dos estilos de condução por viagem e veículo (APP2Car).



Fonte: Elaborado pelo Autor, 2025.

minantemente os estilos “Cauteloso” e “Normal”, com uma ausência notável do estilo “Agressivo” em todas as sessões. Isso indica um padrão de condução consistentemente moderado para este veículo e motorista. Observa-se uma maior proporção do estilo “Normal” em FASTBACK-1 e FASTBACK-2, enquanto FASTBACK-3 e FASTBACK-4 exibem um aumento na proporção do estilo “Cauteloso”.

As viagens da Hilux (HILUX-1 a HILUX-4) caracterizam-se pela exclusividade dos estilos “Cauteloso” e “Normal”, sem registro de comportamento “Agressivo” em nenhuma das sessões. O estilo “Normal” é o mais representativo, especialmente em HILUX-2 e HILUX-3, enquanto HILUX-1 e HILUX-4 mostram uma proporção mais equilibrada entre os estilos “Cauteloso” e “Normal”.

O Polo Branco exibe um padrão misto. POLO BRANCO-1 e POLO BRANCO-2 registram apenas os estilos “Cauteloso” e “Normal”, com uma forte predominância do estilo “Cauteloso” em POLO BRANCO-1. Em POLO BRANCO-3 e POLO BRANCO-4, o estilo “Agressivo” emerge, embora em menor proporção, indicando uma alternância no perfil de condução do motorista.

Para o Polo Prata, as quatro viagens (POLO PRATA-1 a POLO PRATA-4) demonstram a presença dos três estilos de condução, com variações nas proporções. As viagens POLO PRATA-2 e POLO PRATA-3 apresentam maior incidência do estilo “Agressivo” em comparação com POLO PRATA-1 e POLO PRATA-4, onde o estilo “Cauteloso” é mais expressivo. A presença constante do estilo “Agressivo” sugere um comportamento de condução mais dinâmico ou em condições que demandam maior intensidade.

Por fim, o Renegade (RENEGADE-1 a RENEGADE-4) exibe um padrão de condução majoritariamente “Normal” e “Cauteloso”, com uma ausência consistente do estilo “Agressivo” em todas as viagens. Há uma predominância do estilo “Normal”, especialmente em RENEGADE-1 e RENEGADE-4, enquanto RENEGADE-2 e RENEGADE-3

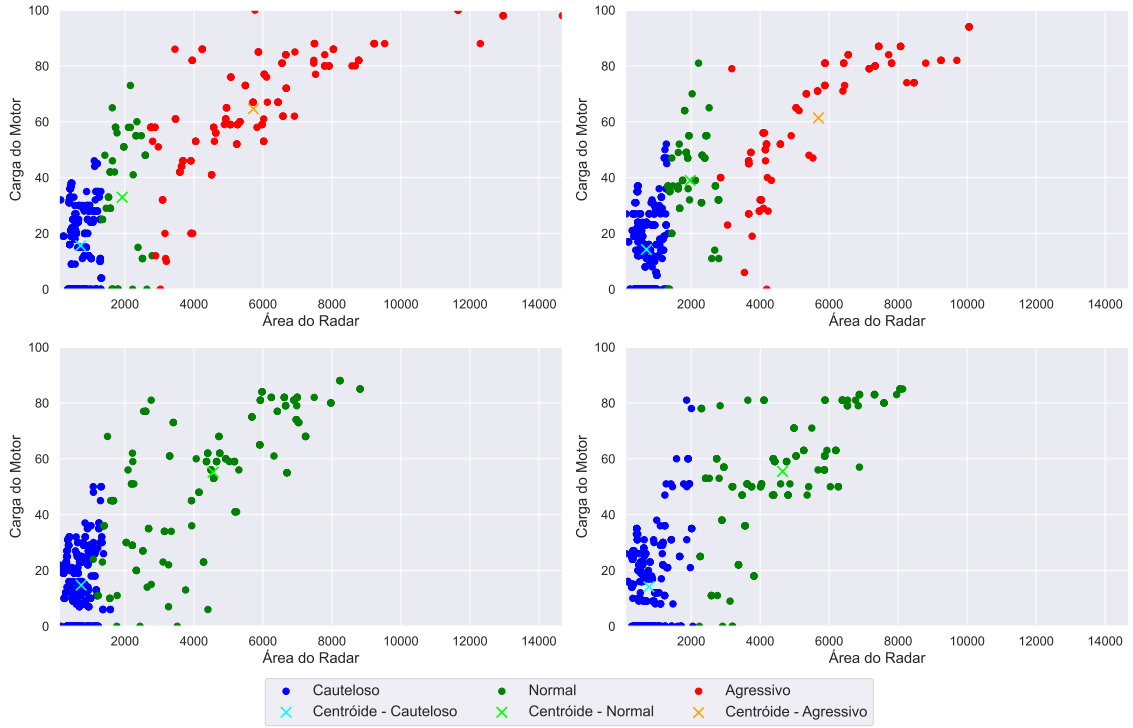
mostram uma distribuição mais equitativa entre os estilos “Cauteloso” e “Normal”.

Em síntese, a análise por viagem revela a capacidade do modelo em capturar nuances no comportamento de condução, que podem ser influenciadas por fatores como o veículo, o motorista e as condições contextuais da viagem. Alguns motoristas mantêm um perfil de condução consistente, enquanto outros demonstram maior variabilidade entre as sessões.

6.2.2 Análise dos Gráficos de Dispersão por Viagem

Esta seção apresenta a análise dos gráficos de dispersão, que ilustram a relação entre a área do radar e a carga do motor para cada viagem, segmentados pelos estilos de condução identificados. Deste modo, observa-se na Figura 6.9 a relação entre a área do radar e a carga do motor ao longo das quatro viagens realizadas. As duas viagens iniciais (Viagem 1 e Viagem 2), onde o perfil “Agressivo” foi identificado (pontos vermelhos), os dados exibem maior dispersão.

Figura 6.9: Relação entre Área do Radar e Carga do Motor - Eclipse Cross (por viagem).



Fonte: Elaborado pelo Autor, 2025.

Em contraste, nas viagens subsequentes (Viagem 3 e Viagem 4), a distribuição concentra-se nos estilos “Normal” (verde) e “Cauteloso” (azul), com menor variabilidade e centróides localizados em regiões de menor intensidade de operação. A ausência de pontos agressivos nas sessões mais recentes sugere uma mudança no padrão de condução, possivelmente influenciada pelo contexto temporal ou por uma adaptação consciente do motorista. A comparação entre as subfiguras evidencia não apenas a transição entre estilos,

mas também a estabilidade dos perfis mais moderados, como indicado pela proximidade dos centróides nas viagens do segundo dia.

Assim, as análises dos gráficos de dispersão, detalhadas no Apêndice A, destacam padrões distintos de estilos de condução para cada veículo. Para o Jeep Renegade e o Fiat Fastback, predominaram os estilos "Cauteloso" e "Normal", com centróides próximos e pouca ou nenhuma manifestação de condução agressiva, indicando um uso moderado e restrito dos veículos. Em contraste, o Volkswagen Polo Branco apresentou o estilo "Agressivo" em uma das viagens, sugerindo uma variação maior dos perfis de condução. Já o Volkswagen Polo Prata demonstrou a presença consistente dos três estilos em todas as viagens, refletindo um comportamento de condução consistente e diversificado. Por fim, a Toyota Hilux manteve padrões estáveis e previsíveis dos estilos "Cauteloso" e "Normal", sem indicativos de agressividade, reforçando um uso contido e consistente.

Além disso, para cada veículo e viagem, as métricas de agrupamento baseadas na área do radar e na carga do motor — como Silhouette, Davies-Bouldin, Calinski-Harabasz e Dunn — foram calculadas. Os resultados, organizados na Tabela 6.3, evidenciam variações entre os padrões de condução.

Para o Eclipse Cross, as métricas de qualidade de *cluster* variam entre as viagens. Na Viagem 1, o Silhouette Score de 0,5348 e o Calinski-Harabasz de 760,49, junto ao Davies-Bouldin de 0,5425, indicam *clusters* com uma definição razoável, embora com certa proximidade, como sugerido pelo baixo Dunn Index de 0,00260. A Viagem 4, por outro lado, apresenta o melhor desempenho: um Silhouette Score de 0,7417 (mais próximo de 1), um Davies-Bouldin de 0,4410 (mais baixo), um Calinski-Harabasz de 1886,47 (mais alto) e um Dunn Index de 0,03230 (mais elevado). Estes valores combinados significam que, para esta viagem, os *clusters* de estilos de condução são mais compactos internamente e bem separados entre si. As Viagens 2 e 3 exibem características intermediárias, com Silhouettes em torno de 0,61 a 0,68 e Calinski-Harabasz acima de 1200, indicando boa qualidade de agrupamento.

No caso do Jeep Renegade, as métricas indicam consistentemente agrupamentos bem definidos para a maioria das viagens. O Silhouette Score para a Viagem 1 é de 0,7038, indicando uma boa coesão interna e separação dos *clusters*. O Davies-Bouldin de 0,5234 complementa essa visão, sugerindo baixa sobreposição entre os grupos, e o Calinski-Harabasz de 1405,81 reforça a densidade e separação. O Dunn Index de 0,00000 para as Viagens 1 e 2 sugere que os *clusters* são muito próximos ou interconectados em seus limites. A Viagem 4 se destaca com um Silhouette de 0,7358, um Davies-Bouldin de 0,4108 (o mais baixo para o Renegade), um Calinski-Harabasz de 2373,48 (o mais alto) e um Dunn Index de 0,01150 (o mais alto), o que implica *clusters* mais compactos e distintamente separados para esta sessão.

Para o Fiat Fastback, observa-se uma variação na qualidade dos *clusters* entre as viagens. A Viagem 4 exibe os resultados mais robustos, com um Silhouette Score de 0,8360 (indicativo de *clusters* muito bem definidos e separados), um Davies-Bouldin de 0,2835 (o mais baixo, sugerindo excelente separação) e um Calinski-Harabasz de 4818,25 (o mais alto, denotando alta densidade e separação). O Dunn Index de 0,03350 para esta viagem também é o maior, confirmando a alta distinção entre os grupos. Em contraste, a Viagem 2 apresenta as métricas menos favoráveis (Silhouette de 0,4966 e Davies-Bouldin

Tabela 6.3: Métricas de Clusterização para Cada Motorista e Veículo.

Veículo	Métrica	Viagem			
		1	**2**	**3**	**4**
Eclipse	Índice de Silhueta	0,5348	0,6148	0,6862	0,7417
	Índice de Davies-Bouldin	0,5425	0,5355	0,5298	0,4410
	Índice de Calinski-Harabasz	760,49	1320,69	1245,39	1886,47
	Índice de Dunn	0,00260	0,00240	0,00080	0,03230
Renegade	Índice de Silhueta	0,7038	0,6410	0,7028	0,7358
	Índice de Davies-Bouldin	0,5234	0,5793	0,4748	0,4108
	Índice de Calinski-Harabasz	1405,81	1035,56	1568,58	2373,48
	Índice de Dunn	0,00000	0,00000	0,00030	0,01150
Fastback	Índice de Silhueta	0,7047	0,4966	0,7728	0,8360
	Índice de Davies-Bouldin	0,4217	0,6814	0,3244	0,2835
	Índice de Calinski-Harabasz	1544,11	554,63	2575,86	4818,25
	Índice de Dunn	0,00009	0,00013	0,00108	0,03350
Polo Branco	Índice de Silhueta	0,7690	0,8086	0,3348	0,6732
	Índice de Davies-Bouldin	0,4354	0,3730	0,7101	0,5459
	Índice de Calinski-Harabasz	1883,18	2511,72	1110,95	790,98
	Índice de Dunn	0,00280	0,07400	0,00000	0,00180
Polo Prata	Índice de Silhueta	0,5800	0,6060	0,5746	0,6024
	Índice de Davies-Bouldin	0,5949	0,5458	0,5304	0,5291
	Índice de Calinski-Harabasz	1360,12	1838,42	1453,37	1606,99
	Índice de Dunn	0,00002	0,00159	0,00187	0,00300
Hilux	Índice de Silhueta	0,6416	0,7374	0,6717	0,7485
	Índice de Davies-Bouldin	0,4620	0,3583	0,4485	0,3603
	Índice de Calinski-Harabasz	1720,01	3044,91	1986,10	3134,82
	Índice de Dunn	0,00004	0,00619	0,00061	0,01969

de 0,6814), o que aponta para agrupamentos com maior sobreposição para aquela sessão específica. As Viagens 1 e 3 mostram um desempenho sólido, com Silhouettes acima de 0,70.

As métricas para o Polo Branco revelam dois cenários distintos. As Viagens 1 e 2 exibem alta qualidade de agrupamento, com Silhouettes acima de 0,76 e Calinski-Harabasz acima de 1880, indicando *clusters* bem formados. A Viagem 2 é notável por seu Silhouette de 0,8086 e um Dunn Index de 0,07400, o maior de todas as viagens analisadas, sugerindo uma separação excepcional entre os *clusters*. Por outro lado, a Viagem 3 apresenta as métricas menos favoráveis: um Silhouette Score de 0,3348 (indicando *clusters* pouco claros ou com sobreposição), um Davies-Bouldin de 0,7101 (o mais alto para este veículo) e um Dunn Index de 0,00000 (sugerindo falta de separação mínima entre os *clusters*). Isso

pode indicar uma maior mistura de comportamentos ou transições mais graduais entre os estilos naquela sessão. A Viagem 4 volta a demonstrar uma boa definição dos *clusters*, com um Silhouette de 0,6732.

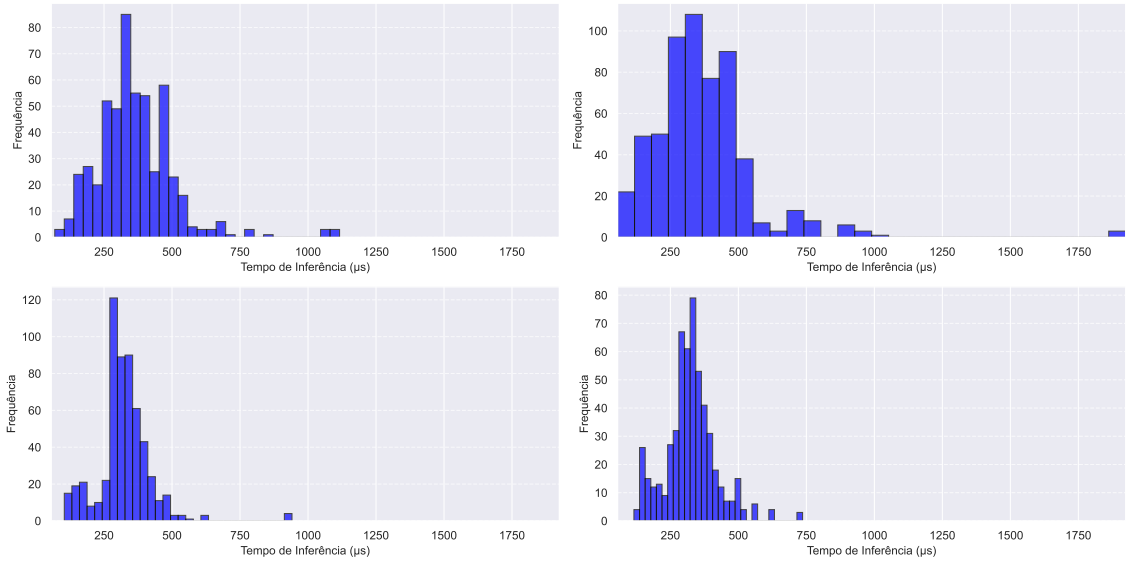
Em relação ao Polo Prata, as métricas de qualidade de cluster mostram uma consistência moderada ao longo das viagens. O Silhouette Score varia entre 0,5746 e 0,6060, o que denota *clusters* com uma separação razoável. O Davies-Bouldin permanece abaixo de 0,60 para todas as sessões, sugerindo que a sobreposição entre os agrupamentos não é excessiva. O Calinski-Harabasz apresenta valores elevados (entre 1360,12 e 1838,42), reforçando a densidade interna dos *clusters*. O Dunn Index, embora numericamente baixo (entre 0,00002 e 0,00300), é consistentemente presente, com a Viagem 4 exibindo o valor mais alto, indicando que os *clusters*, embora talvez próximos, mantêm uma distinção.

Por fim, para a Toyota Hilux, as métricas indicam uma segmentação de *clusters* consistentemente forte. O Silhouette Score é elevado em todas as viagens (entre 0,6416 e 0,7485), sendo a Viagem 4 a que apresenta o maior valor (0,7485). Isso significa que os pontos dentro dos *clusters* estão bem agrupados e distantes dos outros *clusters*. Os valores de Davies-Bouldin são uniformemente baixos (entre 0,3583 e 0,4620), o que é um indicativo de boa separação entre os grupos. O Calinski-Harabasz é consistentemente alto, com picos acima de 3000 nas Viagens 2 e 4, o que reforça a alta densidade e boa separação dos *clusters*. O Dunn Index, embora com valores absolutos pequenos, atinge seu maior valor para a Hilux na Viagem 4 (0,01969), confirmando que os agrupamentos foram os mais compactos e distintamente separados nesta sessão.

Outra análise foi a da distribuição dos tempos de inferência do modelo de classificação de comportamento para cada veículo ao longo das viagens. Para o Eclipse Cross, a Figura 6.10 ilustra essa distribuição, demonstrando que os tempos de execução se mantiveram em níveis adequados para aplicações embarcadas em tempo real, com médias variando entre 322,8 μs e 368,1 μs. Observou-se que as viagens (1 e 2) apresentaram maiores variações e picos de tempo, enquanto as Viagens (3 e 4) revelaram uma distribuição mais concentrada e simétrica, com desvios padrão e valores máximos reduzidos. Essa análise detalhada do 95º percentil reforça a consistência observada nas sessões posteriores.

Para os demais veículos, cujos histogramas detalhados podem ser encontrados no Apêndice A, observou-se padrões de desempenho computacional consistentes entre os veículos, apesar de algumas variações. Para o Jeep Renegade, as distribuições foram predominantemente assimétricas, com a maioria das inferências ocorrendo entre 200 μs e 450 μs, embora a Viagem 3 tenha exibido picos de latência. O Fiat Fastback demonstrou uma melhoria nas viagens posteriores (3 e 4), com médias e dispersões menores em comparação com as viagens iniciais. De forma similar, o Volkswagen Polo Branco demonstrou maior estabilidade e eficiência nas últimas sessões, com tempos médios mais baixos e menor variabilidade. O Volkswagen Polo Prata destacou-se pela consistência geral, com a maioria das inferências abaixo de 670 μs nas últimas viagens, e a Toyota Hilux apresentou uma redução nos tempos médios e na dispersão nas viagens finais. Em síntese, os resultados demonstram que, mesmo com flutuações pontuais, o modelo mantém um desempenho estável e eficiente, com tempos de resposta compatíveis para aplicações embarcadas em tempo real em diversas condições de condução.

Figura 6.10: Distribuição dos tempos de execução do Eclipse Cross no App2Car.



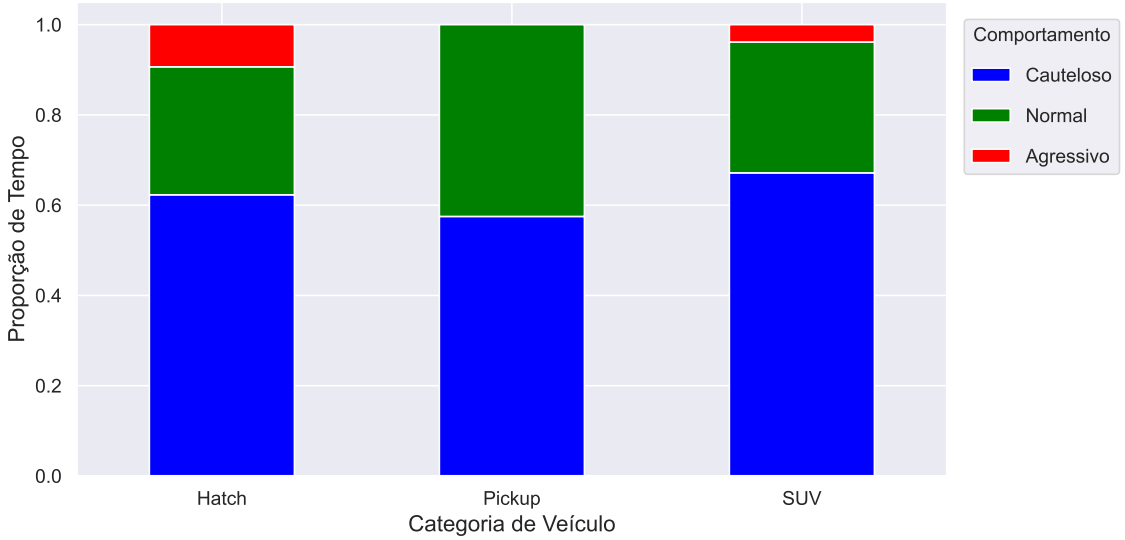
Fonte: Elaborado pelo Autor, 2025.

6.2.3 Análise por Categoria de Veículo

Esta subseção explora como o comportamento de condução varia entre diferentes categorias de veículos (Hatch, SUV e Pickup). Assim, analisando a proporção dos estilos de condução por categoria veicular, como ilustrado na Figura 6.11, observa-se diferenças nos padrões comportamentais. Os veículos Hatch (os dois Polos) apresentam a maior ocorrência do estilo "Agressivo", representando cerca de 9,25% do tempo total de condução. Isso pode sugerir que a agilidade e menor massa desses veículos permitem uma condução mais intensa. Em contraste, os SUVs exibem uma predominância dos estilos "Cauteloso" (aproximadamente 67%) e "Normal", com uma participação menor do perfil "Agressivo" (3,84%). A Pickup (Hilux) se destaca por ter quase a totalidade de sua condução nos estilos "Cauteloso" e "Normal", sem qualquer registro do estilo "Agressivo". Essa ausência reforça a ideia de que o tipo de veículo, além do perfil do condutor, influencia o estilo de condução, limitando manobras mais bruscas. De modo geral, a comparação entre as categorias mostra que o estilo de condução não depende apenas do porte do veículo, mas também do seu uso e da postura do motorista, com Hatches demonstrando maior diversidade e Pickups os padrões mais estáveis.

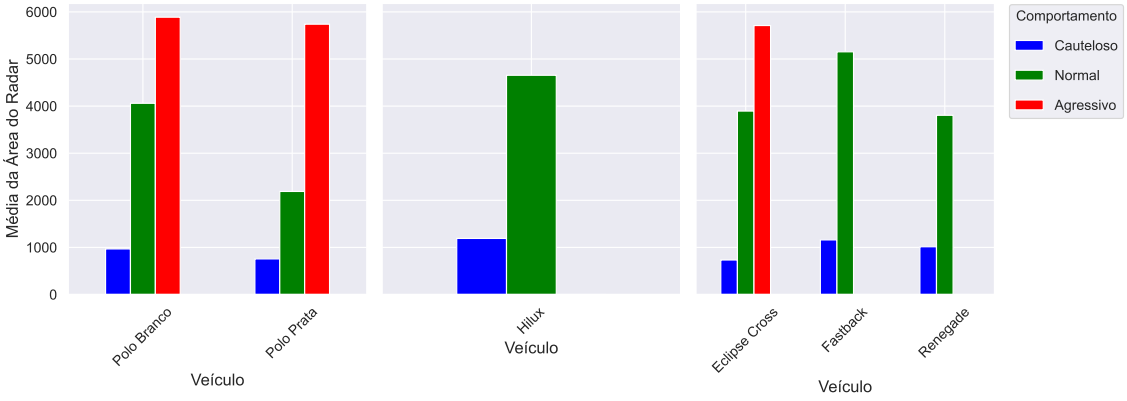
A distribuição da área do radar por estilo de condução e categoria veicular é outro ponto para a análise, e está ilustrada na Figura 6.12. Para os Hatch, percebe-se uma clara distinção o perfil "Agressivo" exibe maiores áreas médias e maior dispersão, indicando uma exploração operacional mais ampla. Já os estilos "Normal" e "Cauteloso" são mais concentrados e separados. Nos SUVs, há uma menor separação entre os estilos, com mais sobreposição, sugerindo transições mais suaves entre os regimes de operação, embora o "Cauteloso" se mantenha focado em valores baixos. Na Pickup, a ausência do estilo "Agressivo" é visível, e os estilos "Cauteloso" e "Normal" mostram distribuições muito próximas e com pouca variabilidade, reforçando a estabilidade operacional desse veículo.

Figura 6.11: Proporção dos estilos de condução por categoria de veículo (APP2Car).



Fonte: Elaborado pelo Autor, 2025.

Figura 6.12: Média da Área do Radar por categoria de veículo (APP2Car).

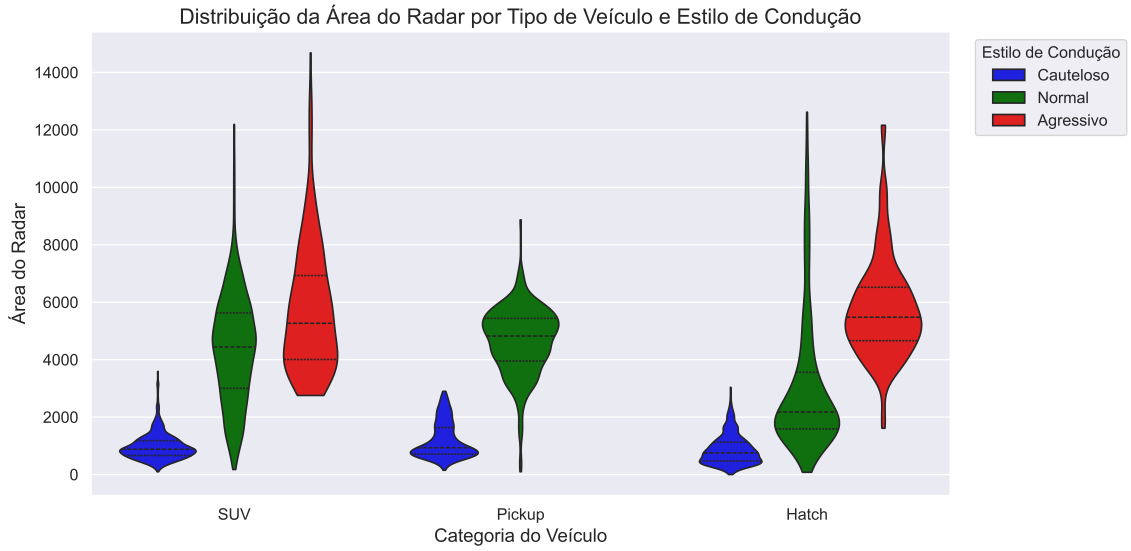


Fonte: Elaborado pelo Autor, 2025.

Deste modo, destaca que os Hatches apresentam maior variação comportamental, SUVs um comportamento intermediário, e Pickups os padrões mais previsíveis. Complementar a essa compreensão, na Figura 6.13, observa-se que, para Hatches, há uma progressão clara nas médias de área do radar entre os estilos, com baixa variabilidade no “Agressivo” e alta no “Normal”. Para SUVs, o padrão crescente também se mantém, mas com maior variabilidade no “Cauteloso”. A Pickup confirma sua estabilidade, com médias elevadas no estilo “Normal” e pouca dispersão.

Os centróides dos estilos de condução, definidos pelas médias da área do radar e da carga do motor para cada veículo e organizados por categoria, são apresentados na Figura 6.14. Nos Hatch, os centróides revelam uma clara separação entre os três estilos de condução. O perfil “Agressivo” ocupa a região de maior área e carga, enquanto o

Figura 6.13: Distribuição da Área do Radar por categoria de veículo e por estilo de condução (APP2Car).



Fonte: Elaborado pelo Autor, 2025.

"Cauteloso" permanece restrito aos menores valores dessas variáveis. A proximidade dos centróides entre os dois veículos Hatch indica uma interpretação homogênea dos estilos. Nos SUVs, há maior dispersão horizontal nos centróides, para classificação "Normal" e "Agressivo", o que reflete variações entre os motoristas na intensidade operacional. A Pickup mostra centróides "Cauteloso" e "Normal" concentrados em uma região estreita, confirmando sua baixa variabilidade comportamental e a ausência do estilo "Agressivo".

Por fim, a distribuição das variáveis operacionais (velocidade, RPM, posição do acelerador e carga do motor) segmentadas por estilo de condução para cada categoria são analisadas. Nos SUVs (Figura 6.15), a posição do acelerador é a variável mais discriminativa, com o "Agressivo" em níveis elevados e o "Cauteloso" em aberturas reduzidas. RPM e carga do motor seguem um padrão similar, enquanto a velocidade mostra sobreposição entre "Normal" e "Agressivo", sugerindo que ela não é um indicador bom para diferenciar estilos em ambientes urbanos.

Para os Hatches (Figura 6.16), a posição do acelerador e a carga do motor também são as mais discriminativas, com o estilo "Agressivo" concentrando-se em altos níveis de utilização, indicando uso intenso e consistente. A RPM acompanha essa tendência, mas a velocidade novamente mostra sobreposição, confirmando que a distinção nos Hatches está mais ligada à forma de condução (aceleração) do que à velocidade final.

Na Pickup (Figura 6.17), observa-se uma alta sobreposição entre os estilos "Cauteloso" e "Normal" em todas as variáveis. As distribuições são estreitas e simétricas, indicando um padrão de condução uniforme e pouca variação entre os perfis. A ausência do estilo "Agressivo" e a rigidez operacional reforçam o perfil estável e previsível do condutor da Hilux, o que pode ser influenciado tanto pelas características do veículo quanto pelo contexto de uso.

Figura 6.14: Centróides por estilo de condução por tipo de veículo (APP2Car).

Fonte: Elaborado pelo Autor, 2025.

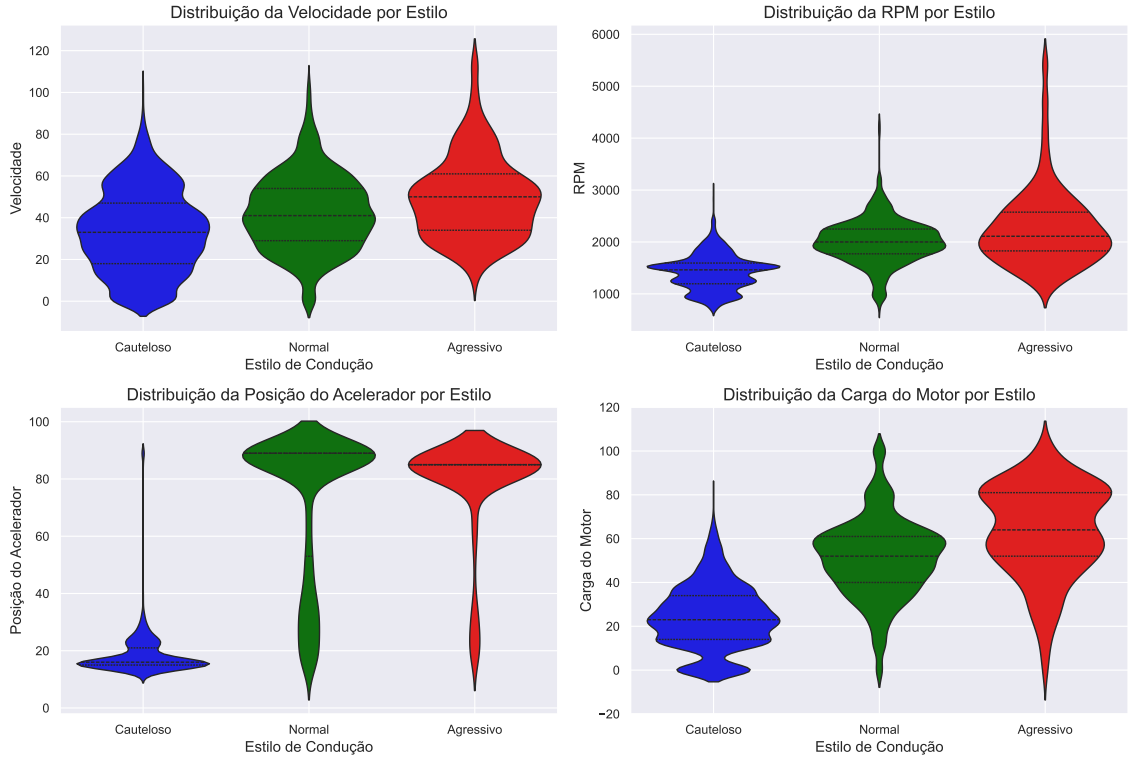
Deste modo, esses resultados destacam que variáveis como posição do acelerador, RPM e carga do motor são eficazes para distinguir os estilos de condução, enquanto a velocidade deve ser interpretada com mais cautela. A análise por categoria demonstra a importância de considerar múltiplas dimensões operacionais para uma classificação precisa dos perfis de condução.

6.3 Discussão

O objetivo central deste trabalho foi investigar uma abordagem para o monitoramento e diagnóstico em tempo real do comportamento do motorista, visando aumentar a segurança veicular, otimizar o consumo de combustível e promover uma condução mais eficiente e sustentável. Para isso, buscou-se responder às questões de pesquisa proposta no estudo de caso. (QP1) como as características individuais dos motoristas e a familiaridade com o veículo influenciam os padrões de comportamento; (QP2) se o tempo médio de permanência nos perfis comportamentais pode ser usado como indicador de estabilidade; (QP3) quais fatores impactam a qualidade dos agrupamentos comportamentais; e (QP4) se o algoritmo MMCloud apresenta desempenho computacional compatível com aplicações em tempo real.

Com relação à QP1, os resultados demonstraram que tanto as características individuais dos motoristas quanto o grau de familiaridade com o veículo exercem influência significativa nos padrões de condução. Por exemplo, um mesmo motorista apresentou estilos distintos ao dirigir veículos diferentes, alternando entre perfis como “Cauteloso”,

Figura 6.15: Distribuição das variáveis operacionais por estilo de condução (SUVs).



Fonte: Elaborado pelo Autor, 2025.

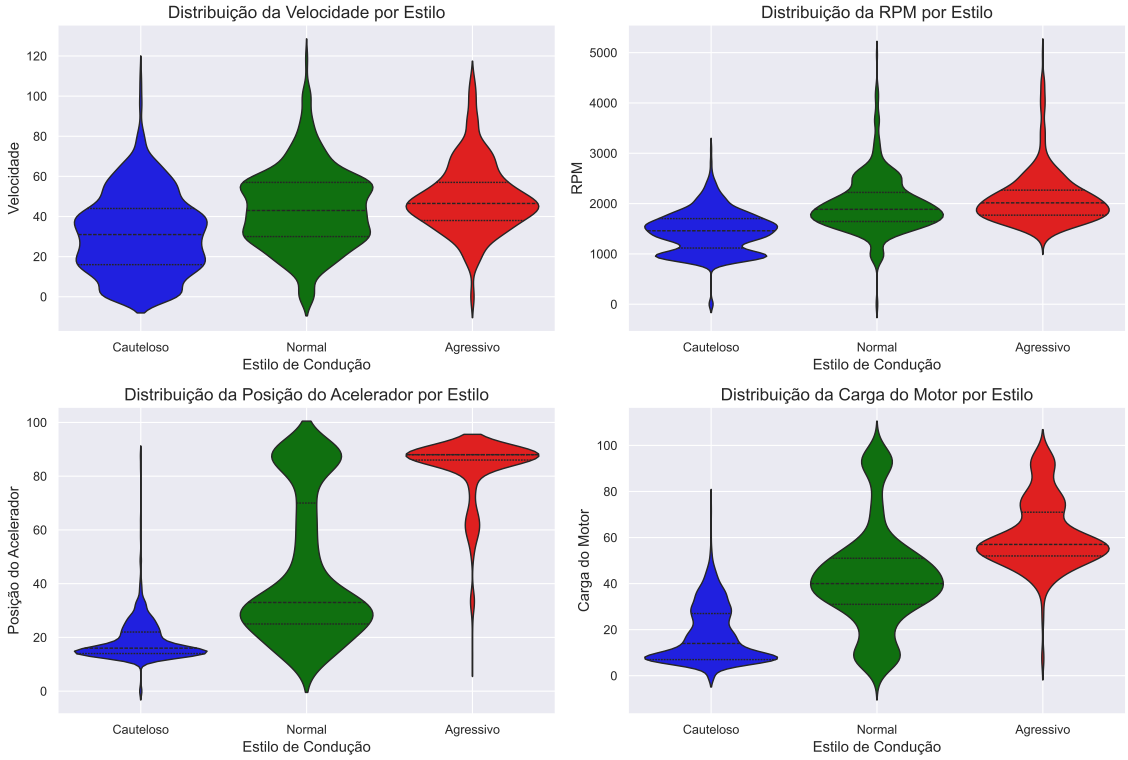
"Normal" e "Agressivo". Esse achado destaca a importância de se considerar o contexto da condução ao avaliar o comportamento, reforçando a necessidade de sistemas de monitoramento adaptativos.

Quanto à QP2, a análise do tempo médio de permanência em cada *cluster* indicou que essa métrica é um bom preditor de estabilidade e previsibilidade no comportamento de direção. Motoristas com maior tempo no perfil "Cauteloso" exibiram estilos de direção mais estáveis e seguros, enquanto aqueles com transições frequentes entre perfis, especialmente para os estilos "Normal" e "Agressivo", apresentaram maior flexibilidade, porém potencial propensão a comportamentos instáveis ou de risco.

Em relação à QP3, a avaliação da qualidade dos agrupamentos revelou que o desempenho do algoritmo MMCloud é influenciado pelas características do motorista e do veículo, bem como pela combinação das variáveis utilizadas na análise. Casos com perfis de direção bem definidos, como o Motorista 2 no Carro 1, resultaram em boa separação dos *clusters*, conforme indicado pelas métricas como o *Silhouette Score* e o *Calinski-Harabasz Index*. No entanto, em contextos de maior sobreposição comportamental, como no Motorista 2 no Carro 2, a qualidade dos agrupamentos foi menor. Isso indica que a seleção ou o enriquecimento das variáveis de entrada pode ser fundamental para melhorar a precisão em contextos mais complexos.

Por fim, respondendo à QP4, os testes de desempenho computacional demonstraram que o algoritmo MMCloud é viável para uso em tempo real, tanto em sistemas embarca-

Figura 6.16: Distribuição das variáveis operacionais por estilo de condução (Hatch).

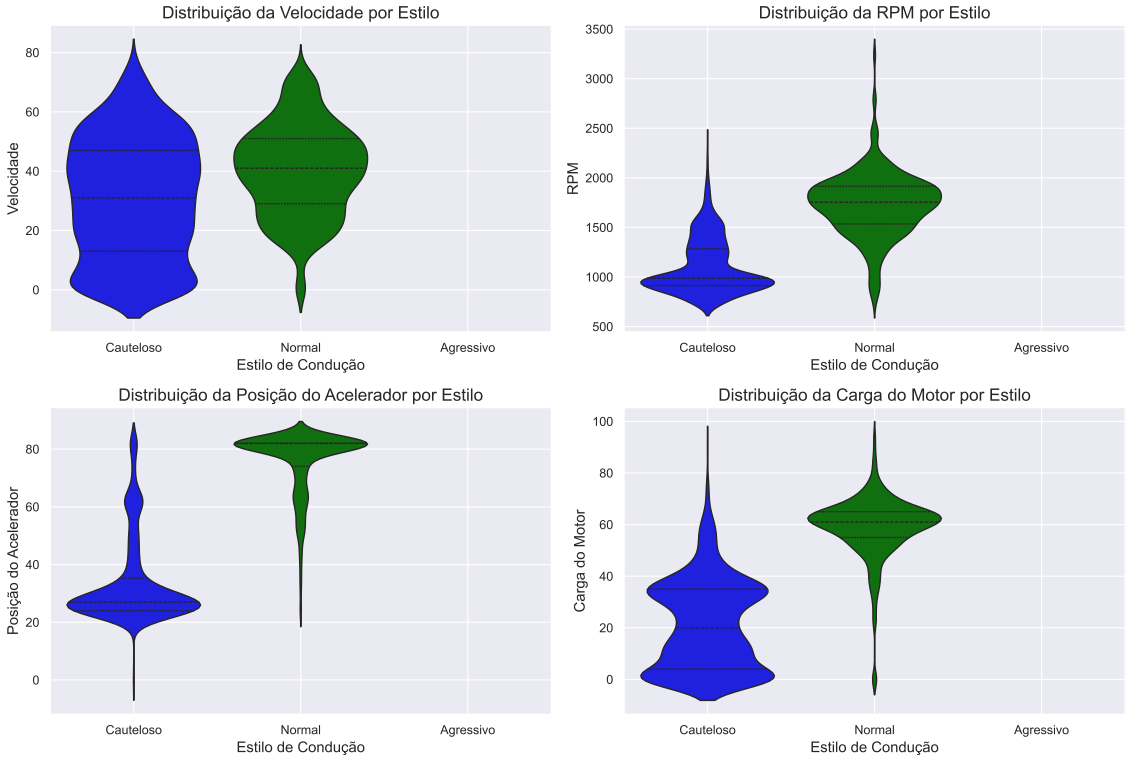


Fonte: Elaborado pelo Autor, 2025.

dos quanto em dispositivos móveis. Os tempos de inferência médios ficaram abaixo de 700 μs nos testes com hardware embarcado, e abaixo de 350 μs na versão implementada no aplicativo *APP2Car*, utilizando comunicação via OBD-II e Bluetooth. Esses resultados comprovam a compatibilidade da solução com aplicações de monitoramento contínuo em contextos comerciais, pessoais e de frotas.

Dessa forma, os resultados deste trabalho evidenciam que a abordagem proposta é eficaz para caracterizar estilos de direção em tempo real, identificar comportamentos de risco, fornecer subsídios para intervenções precoces e contribuir para a redução do consumo de combustível e do desgaste veicular. Além disso, a flexibilidade computacional da solução permite sua integração em plataformas diversas, ampliando seu potencial de aplicação em sistemas de telemetria veicular.

Figura 6.17: Distribuição das variáveis operacionais por estilo de condução (Pickup).



Fonte: Elaborado pelo Autor, 2025.

Seção 7

Conclusão

Este estudo teve como objetivo investigar a viabilidade e a eficácia da abordagem de monitoramento e diagnóstico do comportamento de motoristas em tempo real, com foco na melhoria da segurança veicular, na otimização do consumo de combustível e na promoção de práticas de direção mais eficientes. Ao longo da pesquisa, conseguiu-se não só responder às questões de pesquisa propostas (QP1 a QP4), mas também alcançar os objetivos específicos delineados inicialmente.

A abordagem demonstrou-se eficaz para a análise de padrões de comportamento de motoristas, com resultados consistentes em ambos os estudos de caso — utilizando o Frematics One+ e o APP2Car. Evidenciou-se que tanto as características individuais do motorista quanto as do veículo influenciam diretamente os padrões de condução, respondendo à QP1. Motoristas que apresentaram maior tempo no estilo "Cauteloso" demonstraram um comportamento mais estável e previsível, enquanto a alternância frequente entre os estilos de comportamento indicou maior flexibilidade, mas com possíveis implicações para a estabilidade e a previsibilidade na condução. Essa variação entre os estilos de direção tem um impacto direto não só na segurança veicular, mas também na eficiência do consumo de combustível, uma vez que estilos de condução mais agressivos ou instáveis podem resultar em maior desperdício de combustível e aumento do risco de acidentes.

Em relação à QP2, o estudo confirmou que o tempo médio gasto em cada perfil de direção é um indicativo robusto de estabilidade comportamental. Os motoristas que permaneceram mais tempo no *cluster* "Cauteloso" e no "Normal" demonstraram um comportamento mais previsível e seguro, como observado nas análises por motorista e por categoria de veículo no estudo APP2Car. Por outro lado, motoristas que alternaram rapidamente entre os *clusters*, ou que apresentaram uma proporção significativa do estilo "Agressivo", indicaram um comportamento mais dinâmico ou imprevisível, o que pode comprometer a segurança em determinadas situações de direção.

Para a QP3, a análise das variáveis associadas ao motorista e ao veículo, como a Área do *soft-sensor* e a Carga do Motor, bem como a Posição do Acelerador e a RPM, revelou seu papel crucial na formação dos *clusters* de qualidade e no diagnóstico preciso do comportamento do motorista. As variáveis selecionadas, especialmente aquelas que indicam a carga e a exigência do veículo, mostraram boa correlação com a qualidade do agrupamento dos dados em ambos os estudos. Contudo, a sobreposição observada entre alguns *clusters* em certas viagens ou categorias de veículo, especialmente em situações com maior variabilidade nas características do motorista ou do veículo, sugere que há es-

paço para refinamentos no algoritmo, a fim de melhorar ainda mais a discriminação entre os grupos, possivelmente incorporando mais variáveis discriminativas ou otimizando os parâmetros do algoritmo.

A eficiência computacional do algoritmo MMCloud (QP4) também foi um aspecto fundamental abordado neste estudo. Os resultados obtidos nos testes com o Freematics One+ e, posteriormente, no APP2Car (smartphone), demonstraram que o MMCloud é capaz de realizar os processos de análise de dados em tempo real de forma eficaz, com tempos de execução consistentemente baixos e compatíveis com os requisitos de sistemas embarcados. Esse desempenho computacional é ideal para a aplicação do algoritmo em ambientes de monitoramento de comportamento de motoristas, especialmente em sistemas de assistência ao motorista que demandam alta velocidade de processamento e mínima latência. Como mencionado anteriormente, para garantir a transparência e facilitar futuras colaborações, todo o código desenvolvido para a implementação, incluindo o algoritmo MMCloud, foi disponibilizado publicamente em um repositório no GitHub[1].

Em termos de implicações práticas, os resultados obtidos demonstram que a análise em tempo real do comportamento do motorista, utilizando algoritmos como o MMCloud, possui potencial para promover uma condução mais responsável e segura. A implementação de sistemas que monitoram e diagnosticam o comportamento do motorista pode contribuir para a redução de acidentes, a otimização do consumo de combustível e a melhoria na eficiência do desempenho veicular. Esses avanços têm implicações não apenas para a segurança pública, mas também para a indústria automotiva e para o desenvolvimento de tecnologias avançadas de assistência ao motorista.

Após a apresentação das conclusões do trabalho, as seções subsequentes abordarão perspectivas de trabalhos futuros que complementam o trabalho desenvolvido. Além disso, os artigos aprovados e submetidos.

7.1 Trabalhos Futuros

A proposta desenvolvida neste trabalho poderá ser estendida em diferentes direções. Uma possibilidade consiste na ampliação da base de dados por meio da coleta de um número maior de viagens, incorporando uma variedade mais ampla de veículos e motoristas, bem como a realização de trajetos mais extensos e diversificados. Essa ampliação pode fornecer subsídios para análises comparativas mais robustas, como a investigação de variações nos padrões de condução ao longo do tempo — por exemplo, em diferentes horários do dia ou dias da semana. A partir dessas análises, seria viável implementar mecanismos de alerta em tempo real para detectar desvios em relação ao padrão usual do motorista.

Outra vertente de continuidade envolve o aprimoramento do algoritmo MMCloud. Nesse sentido, sugere-se investigar o uso de métricas de distância alternativas à euclidiana, como a distância de Mahalanobis ou métricas baseadas em densidade, a fim de aumentar a acurácia dos agrupamentos e mitigar a sobreposição entre *clusters*. Além disso, podem ser exploradas estratégias de otimização para redução do tempo de execução do algoritmo, especialmente considerando aplicações embarcadas com restrições de

[1] `https://github.com/conect2ai/MMCloud`

recursos computacionais.

Por fim, destaca-se como perspectiva promissora a integração do algoritmo com modelos de linguagem de pequeno porte (*Small Language Models* – SLMs). Essa integração permitiria a interpretação contextualizada da saída do modelo, viabilizando a geração de recomendações automáticas ao motorista. Entre os possíveis usos, incluem-se sugestões para condução mais econômica e sustentável, alertas de risco em regiões com histórico de acidentes ou infrações, e orientações baseadas em padrões previamente identificados. Tal abordagem pode contribuir para o desenvolvimento de sistemas embarcados mais inteligentes, capazes de combinar diagnóstico comportamental com *feedback* personalizado em tempo real.

7.2 Contribuições

Os artigos publicados a partir do desenvolvimento deste trabalho estão listados abaixo, em ordem cronológica:

Artigo 1: Medeiros, Morsinaldo; Flores, Thommas; Silva, Marianne; Silva, Ivanovitch. **A Multi-Layered Methodology for Driver Behavior Analysis Using TinyML and Edge Computing**. 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), p. 1–8, 2024.

Neste trabalho, foi proposta uma metodologia orientada a fluxos de dados para análise de comportamento de motoristas em tempo real, utilizando computação em borda e TinyML. A abordagem integra sensores virtuais, o *framework* TEDA e um algoritmo de *clustering* incremental para detecção e classificação de padrões comportamentais. Utilizando hardware de baixo consumo energético, o estudo conduzido em Natal-RN, Brasil, validou a eficácia da metodologia em um estudo de caso prático com dois participantes, demonstrando a capacidade agrupar os perfis de condução e de fornecer *insights* para a otimização da eficiência veicular e redução do consumo de combustível.

Artigo 2: Medeiros, M.; Andrade, M.; Medeiros, T.; Silva, M.; and Silva, I. **Anomaly Detection in Simulated Vehicle Dynamics Using BeamNG.tech and the TEDA Framework**. American Workshop on Safe and Secure Vehicles, 2024.

Neste trabalho, apresenta-se uma abordagem que integra a plataforma de simulação BeamNG.tech, um sistema de coleta de dados veiculares e o *framework* TEDA para a detecção de anomalias em dados de velocidade de veículos. Primeiramente, a metodologia envolve a criação de um ambiente de simulação personalizado. Em seguida, aplica-se o TEDA para análise de dados em tempo real, além de garantir a transmissão eficiente dos dados para processamento e armazenamento, utilizando Node-RED e TimescaleDB. Para avaliar a abordagem, foi realizado um estudo de caso em que um motorista percorreu uma rota predefinida, com variações intencionais de velocidade, simulando comportamentos atípicos. Como resultado, o TEDA identificou 1.110 anomalias em 6.388 amostras de velocidade, correspondendo a 17,38% dos dados, o que evidencia a eficácia na detecção de padrões incomuns. Por fim, a integração proposta viabiliza análises em tempo real, com potencial para contribuir com o desenvolvimento de sistemas de assistência ao motorista e tecnologias de segurança automotiva.

Artigo 3: Andrade, M.; Medeiros, M.; Medeiros, T.; Silva, M.; and Silva, I. **Inferring Driver Behavior Profiles Using Digital Twins in Simulated Environments**. American Workshop on Safe and Secure Vehicles, 2024.

Neste artigo, foi proposta uma metodologia para inferir perfis de comportamento de motoristas utilizando *digital twins* em ambientes simulados, especificamente no Euro Truck Simulator 2 (ETS2). A abordagem integra sensores virtuais para coletar dados de telemetria, como velocidade, RPM, aceleração e consumo de combustível, e utiliza a métrica de área do radar para analisar padrões de condução. Um estudo de caso foi conduzido com um motorista que simulou rotas sob duas condições de condução: cautelosa e agressiva. Os resultados demonstraram que a área do radar é eficaz na distinção entre os estilos de condução, com a condução cautelosa apresentando menor variabilidade e maior eficiência no consumo de combustível. A metodologia mostrou-se capaz de capturar informações em tempo real sobre o comportamento do motorista, destacando o potencial dos ambientes simulados e dos digital twins para análises detalhadas de segurança e eficiência no transporte.

Além destes, os artigos publicados em conjunto com o grupo de pesquisa estão listados abaixo, em ordem cronológica:

- Conferência Nacional/Internacional:

  Artigo 1: Azevedo, M.; Andrade, M.; Medeiros, M.; Medeiros, T.; Silva, M.; and Silva, I. **Optimizing Vehicle IoT Systems: SUMO-Digital Twin Performance Analysis**. IEEE MetroInd4.0&IoT (MetroInd), 2024.

  Neste trabalho, analisa-se o desempenho de uma arquitetura de dados veiculares utilizando o simulador SUMO para tráfego urbano. A arquitetura inclui um servidor configurado com Node-RED para coleta e pré-processamento de dados, integrando um banco de dados Timescale para armazenamento de séries temporais. Em três simulações com 91.296 veículos cada, totalizando 273.888 veículos, avaliaram-se métricas de utilização de CPU e memória, taxa de transferência e latência. Os resultados demonstram robustez da arquitetura, com picos de 99,9% de uso de CPU, transferência média de 297 dados por segundo e latência média de 399 ms, validando sua aplicação em sistemas de mobilidade urbana e tecnologias automotivas.

  Artigo 2: Amaral, M.; Medeiros, M.; Andrade, M.; Flores, T.; Silva, M.; and Silva, I. **TinyML Implementation and Optimization for Fuel Type Classification on OBD-II Edge device**. Symposium on Internet of Things (SIoT), 2024.

  Neste trabalho, propõe-se uma abordagem baseada em TinyML para identificar automaticamente o tipo de combustível utilizado por veículos, utilizando dados de sensores veiculares e o algoritmo Random Forest. O modelo foi implementado no dispositivo OBD-II Edge Freematics ONE+, permitindo coleta e análise de dados em tempo real. Um estudo de caso avaliou a eficácia do modelo com veículos como Fiat Mobi e Honda WR-V, obtendo taxas de precisão superiores a 89%. A implementação demonstrou viabilidade em dispositivos com recursos limitados, contribuindo para a gestão ambiental no setor de transporte ao facilitar a criação de inventários de carbono mais precisos e apoiar políticas de redução de emissões.

Artigo 3: Flores, T.; Medeiros, M.; Silva, M.; e Silva, I. **Integrando TinyML em Veículos Flex: Novas Perspectivas para Eficiência Energética e Controle de Poluentes**. Anais da Sociedade Brasileira de Automática, 2024.

Neste trabalho, apresenta-se uma metodologia baseada em TinyML para estimar a eficiência nas rodas por emissão de CO_2 em veículos flex. A abordagem utiliza algoritmos de aprendizado de máquina integrados ao sistema OBD-II para realizar análises em tempo real. Foram comparados três modelos — MLP, Random Forest e Decision Tree — avaliados em termos de eficiência computacional e precisão. Os resultados mostram que a MLP apresentou o menor erro médio absoluto (0,27), enquanto os demais modelos se destacaram pela eficiência em tempo de execução. A metodologia proposta contribui para o avanço da sustentabilidade no transporte rodoviário ao permitir o monitoramento detalhado da eficiência energética e das emissões de poluentes em veículos de forma precisa e em dispositivos com recursos limitados.

Artigo 4: Medeiros, T.; Medeiros, M.; Machado, H.; Alves, G.; Silva, M.; e Silva, I. **Integração de Modelos de Linguagem de Grande Escala com a técnica RAG para a Simplificação de Consultas a Manuais Automotivos**. Anais da Sociedade Brasileira de Automática, 2024.

Neste trabalho, propõe-se uma arquitetura baseada na integração de Modelos de Linguagem de Grande Escala (LLMs) com a técnica *Retrieval-Augmented Generation* (RAG) para simplificar o acesso a informações em manuais automotivos. A abordagem suporta consultas em texto, áudio e imagem, utilizando uma API desenvolvida para interações eficientes. Um estudo de caso avaliou a solução com base em consultas a manuais do Volkswagen Taos 2023, empregando métricas como ROUGE, METEOR e BERTScore para medir a precisão e a qualidade semântica das respostas. Os resultados indicam melhorias significativas na acessibilidade e na rapidez das consultas, destacando o potencial da arquitetura para transformar a interação com manuais complexos e extensos.

Artigo 5: Medeiros, M.; Costa, H.; Silva, M.; e Silva, I. **Avaliação de Algoritmos de Compressão de Séries Temporais Multivariadas com TinyML em Dispositivos Embarcados**. Workshop de Computação Urbana (CoUrb), 2025.

Neste trabalho, avaliam-se dois algoritmos de compressão de séries temporais multivariadas — MPTAC e MSTAC — implementados em dispositivos embarcados com recursos limitados, utilizando o OBD-II Edge Freematics One+ em cenários veiculares reais. O MPTAC obteve melhor fidelidade na reconstrução dos dados, enquanto o MSTAC alcançou maior taxa de compressão. As métricas utilizadas incluíram RMSE, NCC, CR, CF e latência computacional. Os resultados indicam que a escolha do algoritmo deve considerar o equilíbrio entre compressão e fidelidade dos dados, contribuindo para a eficiência energética e redução de transmissão em aplicações IoT automotivas.

- *Journal*:

Artigo 1: Andrade, M.; Medeiros, M.; Medeiros, T.; Amaral, M.; Silva, M.;

and Silva, I. **On the Use of Biofuels for Cleaner Cities: Assessing Vehicular Pollution through Digital Twins and Machine Learning Algorithms**, Sustainability 16 (2), 708, 2024.

Neste trabalho, apresenta-se uma metodologia para estimar emissões de CO_2 em veículos flex, integrando sensores OBD-II, aprendizado de máquina e gêmeos digitais. A abordagem utiliza sensores MAF e AFR para estimar emissões em tempo real, complementada pelo algoritmo TEDA para identificação de *outliers*. Adicionalmente, cenários simulados no SUMO replicam condições reais para análises extensivas. Os resultados demonstram emissões de CO_2 significativamente menores para etanol em comparação com gasolina, destacando o impacto ambiental positivo de combustíveis sustentáveis. A metodologia proposta contribui para políticas de transporte mais limpas e promove a transição energética no setor automotivo.

Artigo 2: Andrade, P.; Silva, M.; Medeiros, M.; Costa, D.G.; and Silva, I.;**TEDA-RLS: A TinyML Incremental Learning Approach for Outlier Detection and Correction**. IEEE Sensors Journal, 2024.

Neste trabalho, propõe-se o algoritmo TEDA-RLS, uma abordagem incremental baseada em TinyML para detecção e correção de *outliers* em fluxos de dados. A metodologia combina o algoritmo TEDA para identificação de *outliers* com o filtro RLS para correção, eliminando a necessidade de treinamento prévio. Como prova de conceito, o algoritmo foi implementado em um scanner OBD-II, realizando coleta e análise de dados veiculares em tempo real. Experimentos com dados simulados e reais demonstraram baixos tempos de processamento e eficiência na detecção e correção de *outliers*, evidenciando a adequação do método para dispositivos de recursos limitados. A proposta contribui para aplicações em contextos automotivos e IoT, destacando-se pela eficiência e adaptabilidade.

Artigo 3: Costa, D.G.; Silva, I.; Medeiros, M.; Bittencourt, J.C.N.; and Andrade, M.; **A method to promote safe cycling powered by large language models and AI agents**. MethodsX, 2024.

Neste trabalho, apresenta-se o método SafeCycle-Assist, que combina Modelos de Linguagem de Grande Escala (LLMs) e agentes baseados em IA para promover a segurança no ciclismo urbano. A abordagem utiliza dados geoespaciais de infraestrutura cicloviária e níveis de risco urbano, processados em três etapas principais: ingestão e preparação de dados, orquestração de agentes e execução de decisões. O método permite que usuários façam consultas textuais ou de áudio sobre a segurança de trajetos, recebendo respostas detalhadas baseadas em dados do OpenStreetMap e da ferramenta CityZones. Testes realizados em Porto, Portugal, demonstraram a eficácia da metodologia, que é facilmente reproduzível para outras cidades, contribuindo para a mobilidade sustentável e o planejamento urbano seguro.

Artigo 4: Flores, Thommas; Medeiros, Morsinaldo; Silva, Marianne; Costa, Daniel G.; and Silva, Ivanovitch. **Enhanced Vector Quantization for Embedded Machine Learning: A Post-Training Approach with Incremental Clustering**. IEEE Access, 2024.

Neste trabalho, foi proposta uma abordagem para otimização da quantização pós-treinamento (PTQ, do inglês *Post-Training Quantization*) em dispositivos embarcados, integrando quantização vetorial (VQ, do inglês *Vector Quantization*) e agrupamento incremental via algoritmo AutoCloud K-Fixed. A técnica foi avaliada com dados automotivos para prever emissões de CO_2, demonstrando compressão superior a 90% com impacto mínimo na precisão. O método foi implementado no hardware Macchina A0, validando sua aplicação em sistemas com recursos limitados.

Artigo 5: Flores, T.K.S.; Silva, I.; Azevedo, M.B.; Medeiros, T.A.; Medeiros, M.A.; Costa, D.G.; Ferrari, P.; and Sisinni, E. **Advancing Tiny Machine Learning Operations: Robust Model Updates in the Internet of Intelligent Vehicles**. IEEE Micro, 2025.

Este trabalho propõe uma metodologia de operações TinyML (TinyMLOps) voltada ao contexto da Internet de Veículos Inteligentes (IoIV), focando na atualização robusta de modelos embarcados. A solução emprega dois dispositivos ESP32 — uma estação de rádio e um *scanner* OBD-II — para transmitir modelos quantizados via protocolo ESP-NOW. O estudo contempla desde a coleta e preparação dos dados até a compressão e atualização remota dos modelos, abordando desafios como concept drift e restrições de memória e energia. Resultados experimentais mostram melhora significativa no desempenho após a atualização do modelo, com baixos tempos de latência e consumo energético moderado, validando a eficácia da abordagem em cenários automotivos reais.

Artigo 6: Andrade, P.; Medeiros, M.; Silva, M.; Costa, D.G.; and Silva, I. **TEDA-Forecasting: An Unsupervised TinyML Incremental Learning Approach for Outlier Processing and Forecasting**. Computing, 2025.

Neste artigo, é proposto o algoritmo TEDA-Forecasting, uma técnica de aprendizado incremental baseada em séries temporais, voltada para execução em dispositivos embarcados com recursos limitados. A metodologia integra o framework TEDA para detecção de outliers com o filtro adaptativo RLS, possibilitando correção em tempo real e previsão de múltiplos valores futuros. A abordagem é validada tanto com dados públicos quanto com dados veiculares coletados por um dispositivo OBD-II Freematics ONE+. Os resultados demonstram elevada acurácia, baixo consumo energético e adequação ao paradigma TinyML, evidenciando o potencial do algoritmo para aplicações em ambientes restritos como sistemas embarcados veiculares.

7.3 Artigos Submetidos

Os artigos submetidos que foram desenvolvidos a partir deste trabalho estão listados abaixo:

Artigo 1: Medeiros, Morsinaldo; Silva, Marianne; Silva, Ivanovitch. **Evolving Online Clustering for Real-Time Driver Behavior Analysis in IoT-Enabled Vehicles**. Evolving Systems, p. 1–15, 2025.

Neste trabalho, propõe-se uma abordagem de clusterização incremental para diagnóstico em tempo real do comportamento de motoristas, utilizando TinyML e sensores virtuais. O sistema integra o algoritmo MMCloud com computação em borda e dados do OBD-II para classificar estilos de condução como cauteloso, normal e agressivo. Validado com dois motoristas em diferentes veículos em Natal-RN, o estudo demonstrou a capacidade do modelo em se adaptar a padrões comportamentais em evolução, contribuindo para práticas de condução mais seguras e eficientes.

Artigo 2: Moraes, Rejanio; Medeiros, Morsinaldo; Milomem, Fellipe; Silva, Marianne; Silva, Ivanovitch. **Arquitetura Embarcada com TinyML e Modelos Linguísticos para Monitoramento Veicular Inteligente**. XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT), p. 1–8, 2025.

Este trabalho propõe uma arquitetura embarcada baseada em Raspberry Pi que integra dados veiculares via OBD-II, GPS e acelerômetro a um pipeline de inferência contextual com TinyML e modelos de linguagem (SLMs). A solução estima estilo de condução, emissão de CO_2, tipo de via e eventos anômalos, gerando alertas interpretáveis em tempo real com um agente linguístico. Um estudo de caso demonstrou a viabilidade técnica e responsividade da abordagem em tráfego urbano e rodoviário em Natal-RN, mesmo sob condições operacionais restritivas.

Referências Bibliográficas

Abernathy, Amber & M Emre Celebi (2022), 'The incremental online k-means clustering algorithm and its application to color quantization', *Expert Systems with Applications* **207**, 117927.

Alalwany, Easa & Imad Mahgoub (2024), 'Security and trust management in the internet of vehicles (iov): Challenges and machine learning solutions', *Sensors* **24**(2), 368.

Almeida, Nayron Morais, Murilo Osorio Camargos, Denis GB Mariano, Carlos HM Bomfim, Reinaldo M Palhares & Walmir M Caminhas (2025), 'An evolving approach to the similarity-based modeling for online clustering in non-stationary environments', *Evolving Systems* **16**(1), 16.

Amutha, AL, R Annie Uthra, J Preetha Roselyn & R Golda Brunet (2021), 'Anomaly detection in multivariate streaming pmu data using density estimation technique in wide area monitoring system', *Expert Systems with Applications* **175**, 114865.

Andrade, Pedro, Ivanovitch Silva, Marianne Silva, Thommas Flores, Jordão Cassiano & Daniel G Costa (2022), 'A tinyml soft-sensor approach for low-cost detection and monitoring of vehicular emissions', *Sensors* **22**(10), 3838.

Angelova, Milena, Veselka Boeva & Shahrooz Abghari (2024), Edgecluster: A resource-aware evolving clustering for streaming data, *em* '2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)', IEEE, pp. 1–10.

Ashenden, Stephanie K (2021), *The era of artificial intelligence, machine learning, and data science in the pharmaceutical industry*, Academic Press.

Ashfahani, Andri & Mahardhika Pratama (2022), 'Unsupervised continual learning in streaming environments', *IEEE transactions on neural networks and learning systems* **34**(12), 9992–10003.

Ba, Wei & Zongquan Gu (2024), 'A novel move-split-merge based fuzzy c-means algorithm for clustering time series', *Evolving Systems* **15**(6), 2273–2295.

Bonaccorso, Giuseppe (2019), *Hands-on unsupervised learning with Python: implement machine learning and deep learning models using Scikit-Learn, TensorFlow, and more*, Packt Publishing Ltd.

Bouhsissin, Soukaina, Nawal Sael & Faouzia Benabbou (2023), 'Driver behavior classification: A systematic literature review', *IEEE Access* .

Caliński, Tadeusz & Jerzy Harabasz (1974), 'A dendrite method for cluster analysis', *Communications in Statistics-theory and Methods* **3**(1), 1–27.

Cao, Keyan, Yefan Liu, Gongjie Meng & Qimeng Sun (2020), 'An overview on edge computing research', *IEEE access* **8**, 85714–85728.

Cesarone, Francesco, Rosella Giacometti & Jacopo Maria Ricci (2024), 'Outlier detection of multivariate data via the maximization of the cumulant generating function', *Journal of Computational and Applied Mathematics* p. 116457.

Conect2AI (2025), 'Conect2ai: Artificial intelligence solutions'. Accessed: 2025-01-15. **URL:** *https://conect2ai.dca.ufrn.br/*

Da Silva, Leonardo Enzo Brito, Niklas Max Melton & Donald C Wunsch (2020), 'Incremental cluster validity indices for online learning of hard partitions: Extensions and comparative study', *IEEE Access* **8**, 22025–22047.

Davies, David L & Donald W Bouldin (1979), 'A cluster separation measure', *IEEE transactions on pattern analysis and machine intelligence* (2), 224–227.

Din, Ikram Ud, Ahmad Almogren & Joel JPC Rodrigues (2024), 'Aiot integration in autonomous vehicles: Enhancing road cooperation and traffic management', *IEEE Internet of Things Journal* .

Dresch, Aline, Daniel Pacheco Lacerda & José Antônio Valle Antunes Jr (2020), *Design Science Research: Método de Pesquisa para Avanço da Ciência e Tecnologia*, 1ª edição, Bookman.

Dunn, Joseph C (1973), 'A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters'.

Fadhil, Jawaher Abdulwahab & Qusay Idrees Sarhan (2020), Internet of vehicles (iov): A survey of challenges and solutions, *em* '2020 21st International Arab Conference on Information Technology (ACIT)', IEEE, pp. 1–10.

Fang, Lin, Hongjie Li, Yingchao Zheng & Xinggang Luo (2025), 'Driving behavior recognition and fuel economy evaluation for heavy-duty vehicles', *Research in Transportation Business & Management* **60**, 101371.

Flores, Thommas, Marianne Silva, Mariana Azevedo, Thais Medeiros, Morsinaldo Medeiros, Ivanovitch Silva, Max Mauro Dias Santos & Daniel G Costa (2023), Tinyml for safe driving: The use of embedded machine learning for detecting driver distraction, *em* '2023 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)', IEEE, pp. 62–66.

Freematics (2025), 'Freematics one+ model h'. Accessed: 2025-01-15. **URL:** *https://freematics.com/pages/products/freematics-one-plus-model-h/*

Gan, Guojun, Chaoqun Ma & Jianhong Wu (2020), *Data clustering: theory, algorithms, and applications*, SIAM.

Géron, Aurélien (2022), *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*, "O'Reilly Media, Inc.".

Gil, Antonio Carlos (2022), *Como elaborar projetos de pesquisa*, 7ª edição, Atlas.

Gorrab, Siwar, Fahmi Ben Rejab & Kaouther Nouira (2024), 'Split incremental clustering algorithm of mixed data stream', *Progress in Artificial Intelligence* **13**(1), 51–64.

Govers, Wim, Aras Yurtman, Turgay Aslandere, Nicole Eikelenberg, Wannes Meert & Jesse Davis (2023), 'Time-shifted transformers for driver identification using vehicle data', *IEEE Transactions on Intelligent Transportation Systems* **25**(5), 3767–3776.

Guan, Tian, Yi Han, Nan Kang, Ningye Tang, Xu Chen & Shu Wang (2022), 'An overview of vehicular cybersecurity for intelligent connected vehicles', *Sustainability* **14**(9), 5211.

Hinder, Fabian, Valerie Vaquet, Johannes Brinkrolf & Barbara Hammer (2023), 'Model-based explanations of concept drift', *Neurocomputing* **555**, 126640.

Hoi, Steven CH, Doyen Sahoo, Jing Lu & Peilin Zhao (2021), 'Online learning: A comprehensive survey', *Neurocomputing* **459**, 249–289.

Huyen, Chip (2022), *Designing machine learning systems*, "O'Reilly Media, Inc.".

Iodice, Gian Marco & Ronan Naughton (2022), *TinyML Cookbook: Combine artificial intelligence and ultra-low-power embedded devices to make the world smarter*, Packt Publishing Ltd.

Kennedy, Brett (2025), *Outlier Detection in Python*, 1ª edição, Manning Publications, Shelter Island, NY, USA.

Kezia, M & KV Anusuya (2023), Iot-based web-integrated obd-ii telematics for real-time vehicle health monitoring system, *em* '2023 International Conference on Intelligent Technologies for Sustainable Electric and Communications Systems (iTech SECOM)', IEEE, pp. 91–96.

Kumar, Raman & Anuj Jain (2023), 'Driving behavior analysis and classification by vehicle obd data using machine learning', *The Journal of Supercomputing* **79**(16), 18800–18819.

Lattanzi, Emanuele & Valerio Freschi (2021), 'Machine learning techniques to identify unsafe driving behavior by means of in-vehicle sensor data', *Expert Systems with Applications* **176**, 114818.

Ma, Ning, Kaijun Wu, Yubin Yuan, Jiawei Li & Xiaoqiang Wu (2025), 'Pmwfcm: A possibility based multikernel weighted fuzzy clustering algorithm for classification of driving behaviors', *Alexandria Engineering Journal* **113**, 249–261.

Marrero, Lester, Daniel Sbárbaro & Luis García-Santander (2024), 'Online demand response characterization based on variability in customer behavior', *Journal of Modern Power Systems and Clean Energy* .

Medeiros, Morsinaldo, Thommas Flores, Marianne Silva & Ivanovitch Silva (2024), A multi-layered methodology for driver behavior analysis using tinyml and edge computing, *em* '2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)', IEEE, pp. 1–8.

Medeiros, Thaís, Miguel Amaral, Matheus Targino, Marianne Silva, Ivanovitch Silva, Emiliano Sisinni & Paolo Ferrari (2023), Tinyml custom ai algorithms for low-power iot data compression: A bridge monitoring case study, *em* '2023 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0&IoT)', IEEE, pp. 54–59.

Mitchell, Tom M. (1997), *Machine Learning*, McGraw-Hill, New York.

Mohammadnazar, Amin, Zulqarnain H Khattak & Asad J Khattak (2024), 'Assessing driving behavior influence on fuel efficiency using machine-learning and drive-cycle simulations', *Transportation Research Part D: Transport and Environment* **126**, 104025.

Montgomery, Douglas C & George C Runger (2020), *Applied statistics and probability for engineers*, John wiley & sons.

Neumann, Tomasz (2024), 'Analysis of advanced driver-assistance systems for safe and comfortable driving of motor vehicles', *Sensors* **24**(19), 6223.

Nordahl, Christian, Veselka Boeva, Håkan Grahn & Marie Persson Netz (2022), 'Evolve-cluster: an evolutionary clustering algorithm for streaming data', *Evolving Systems* **13**(4), 603–623.

Parisi, German I, Ronald Kemker, Jose L Part, Christopher Kanan & Stefan Wermter (2019), 'Continual lifelong learning with neural networks: A review', *Neural networks* **113**, 54–71.

Patel, Ankur A (2019), *Hands-on unsupervised learning using Python: how to build applied machine learning solutions from unlabeled data*, O'Reilly Media.

Raschka, Sebastian (2024), *Build a Large Language Model (From Scratch)*, Manning Publications Co.

Ren, Haoyu, Darko Anicic & Thomas A Runkler (2021), Tinyol: Tinyml with online-learning on microcontrollers, *em* '2021 international joint conference on neural networks (IJCNN)', IEEE, pp. 1–8.

Rousseeuw, Peter J (1987), 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of computational and applied mathematics* **20**, 53–65.

Signoretti, Gabriel, Marianne Silva, Pedro Andrade, Ivanovitch Silva, Emiliano Sisinni & Paolo Ferrari (2021), 'An evolving tinyml compression algorithm for iot environments based on data eccentricity', *Sensors* **21**(12), 4153.

Silva, Marianne Batista Diniz da (2022), 'Uma metodologia orientada a fluxo de dados para modelagem do comportamento de motoristas'.

Silva, Marianne, Thais Medeiros, Mariana Azevedo, Morsinaldo Medeiros, Mikael Themoteo, Tatiane Gois, Ivanovitch Silva & Daniel G Costa (2023), An adaptive tinyml unsupervised online learning algorithm for driver behavior analysis, *em* '2023 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)', IEEE, pp. 199–204.

Situnayake, Daniel & Jenny Plunkett (2023), *AI at the Edge*, "O'Reilly Media, Inc.".

Soumaya, OUNACER, TALHAOUI Mohamed Amine, Ardchird Soufiane, Daif Abderrahmane & Azouazi Mohamed (2017), 'Real-time data stream processing challenges and perspectives', *International Journal of Computer Science Issues (IJCSI)* **14**(5), 6–12.

Syahputri, Ziana, Machrani Adi Putri Siregar et al. (2024), 'Determining the optimal number of k-means clusters using the calinski harabasz index and krzanowski and lai index methods for groupsing flood prone areas in north sumatra', *Sinkron: jurnal dan penelitian teknik informatika* **8**(1), 571–580.

Tripathy, BK, Anveshrithaa Sundareswaran & Shrusti Ghela (2021), *Unsupervised learning approaches for dimensionality reduction and data visualization*, CRC Press.

Valente, Jorge, Cláudia Ramalho, Pedro Vinha, Carlos Mora & Sandra Jardim (2024), 'Using machine learning to understand driving behavior patterns', *Procedia Computer Science* **239**, 1823–1830.

Verdhan, Vaibhav (2025), *Data Without Labels: Master unsupervised learning with deep learning and generative AI*, Manning Publications. MEAP version began November 2020, estimated publication Summer 2025.

Warden, Pete & Daniel Situnayake (2019), *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*, O'Reilly Media.

Wazlawick, Raul Sidnei (2020), *Metodologia de pesquisa para ciência da computação*, 3ª edição, GEN LTC.

Wei, Chongbo, Gaogang Xie & Zulong Diao (2023), 'A lightweight deep learning framework for botnet detecting at the iot edge', *Computers & Security* **129**, 103195.

Xie, Bingyan, Tiezhu Li, Tianhao Liu, Haibo Chen, Hu Li & Ying Li (2024), 'Exploring high-emission driving behaviors of heavy-duty diesel vehicles based on engine principles under different road grade levels', *Science of The Total Environment* **951**, 175443.

Yan, Huigui, Jiale Liu, Da Han, Dianlong You, Hongtao Wu, Zhen Chen, Xianshan Li, Shunfu Jin & Xindong Wu (2025), 'Online learning from incomplete data streams with partial labels for multi-classification', *Information Sciences* **689**, 121411.

Yang, Li & Abdallah Shami (2022), 'Iot data analytics in dynamic environments: From an automated machine learning perspective', *Engineering Applications of Artificial Intelligence* **116**, 105366.

Yarlagadda, Jahnavi, Pranjal Jain & Digvijay S Pawar (2021), 'Assessing safety critical driving patterns of heavy passenger vehicle drivers using instrumented vehicle data–an unsupervised approach', *Accident Analysis & Prevention* **163**, 106464.

Yen, Meng-Hua, Shang-Lin Tian, Yan-Ting Lin, Cheng-Wei Yang & Chi-Chun Chen (2021), 'Combining a universal obd-ii module with deep learning to develop an eco-driving analysis system', *Applied Sciences* **11**(10), 4481.

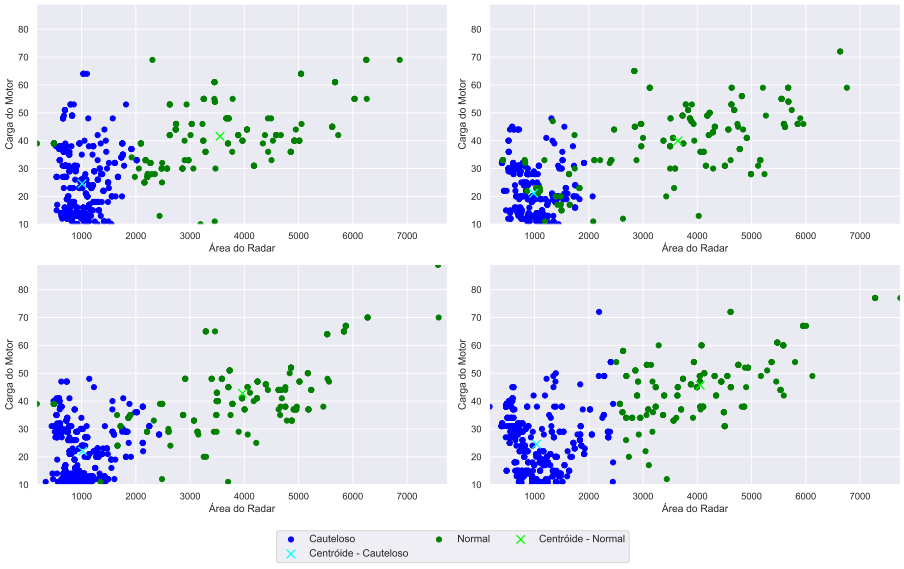
Zhang, Si-si, Jian-wei Liu & Xin Zuo (2021), 'Adaptive online incremental learning for evolving data streams', *Applied Soft Computing* **105**, 107255.

Apêndice A

Distribuição dos Tempos de Inferência dos veículos por Viagem

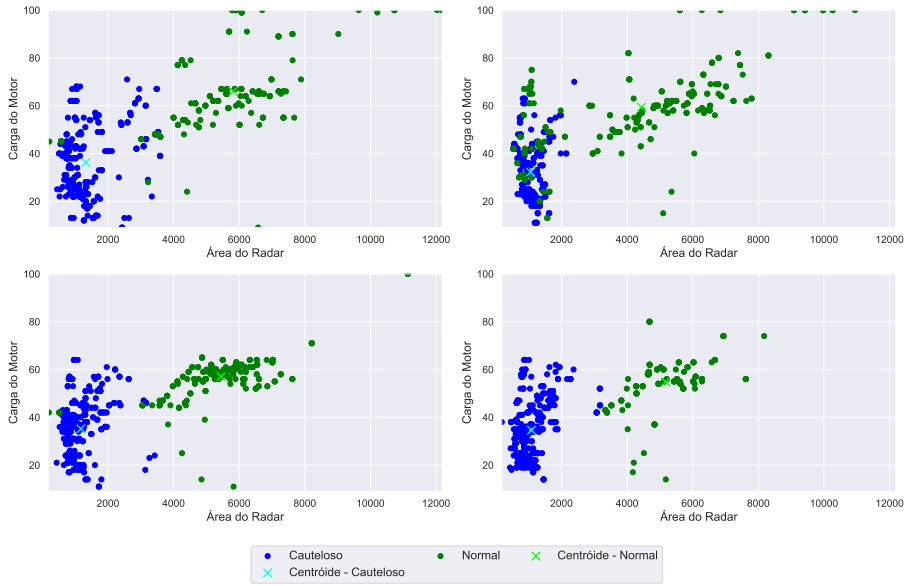
Conforme detalhado na Seção 6, apresentam-se as figuras de distribuição dos tempos de inferência para os veículos Jeep Renegade, Fiat Fastback, Volkswagen Polo (Branco), Volkswagen Polo (Prata) e Toyota Hilux, segmentadas por viagem. Essa visualização oferece uma análise detalhada do comportamento de condução em cada viagem.

Figura A.1: Gráfico de dispersão do Jeep Renegade por viagem.



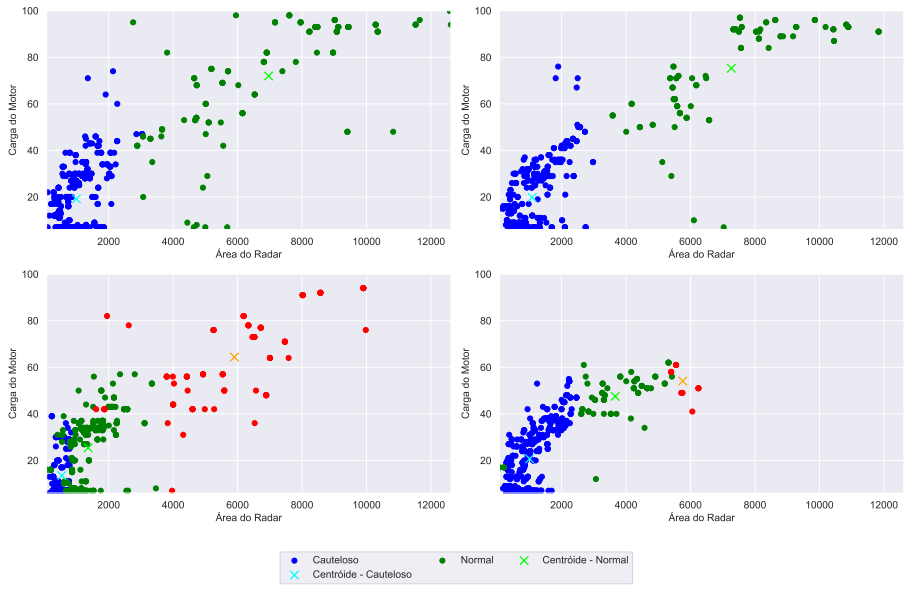
Fonte: Elaborado pelo Autor, 2025.

Figura A.2: Gráfico de dispersão do Fiat Fastback por viagem.



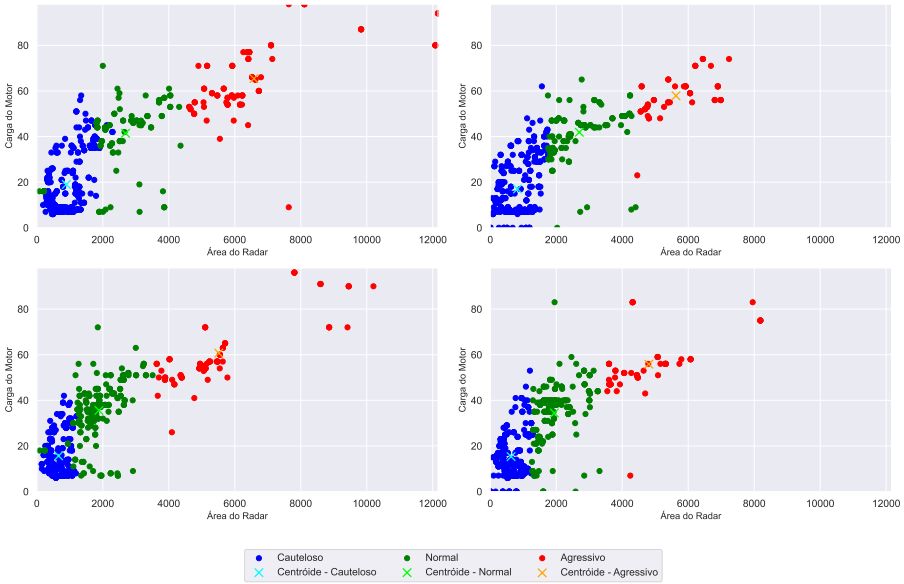
Fonte: Elaborado pelo Autor, 2025.

Figura A.3: Gráfico de dispersão do Volkswagen Polo (Branco) por viagem.



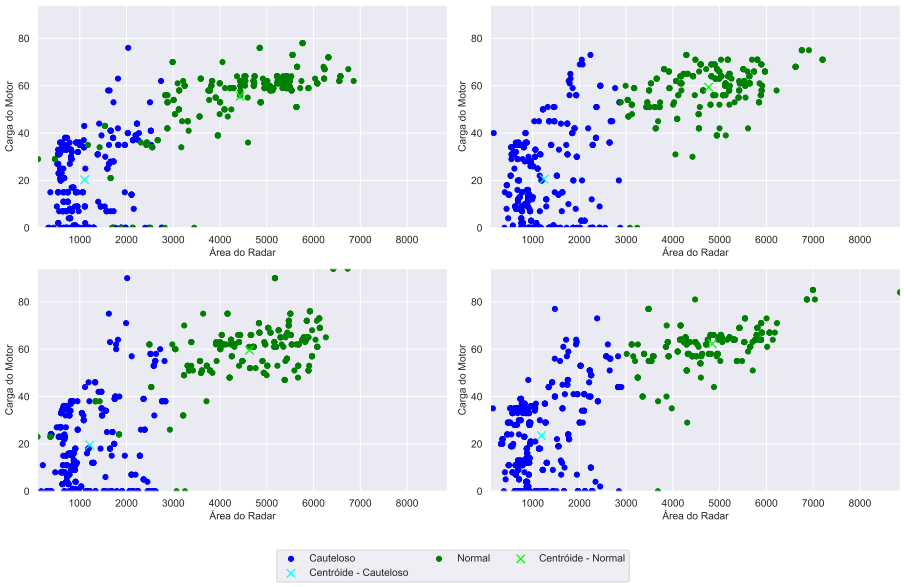
Fonte: Elaborado pelo Autor, 2025.

Figura A.4: Gráfico de dispersão do Volkswagen Polo (Prata) por viagem.



Fonte: Elaborado pelo Autor, 2025.

Figura A.5: Gráfico de dispersão do Toyota Hilux por viagem.



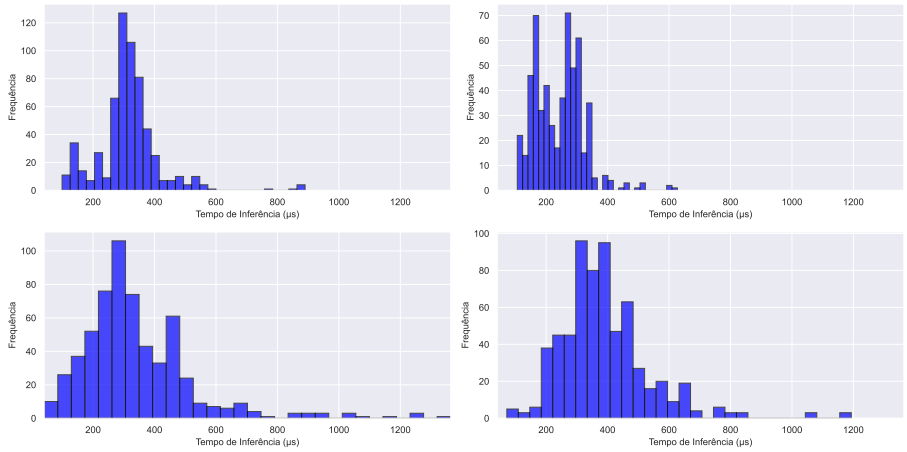
Fonte: Elaborado pelo Autor, 2025.

Apêndice B

Tempos de Inferência dos Veículos por por Viagem

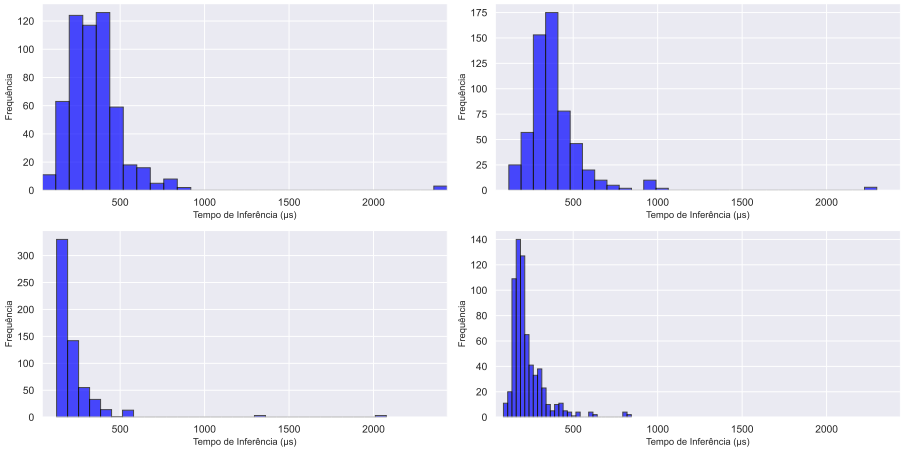
Complementando a discussão da Seção 6, os histogramas dos tempos de inferência para os veículos Jeep Renegade, Fiat Fastback, Volkswagen Polo (Branco e Prata) e Toyota Hilux evidenciam como as latências se distribuem em cada viagem, proporcionando uma anális do desempenho do modelo.

Figura B.1: Distribuição dos tempos de execução do Jeep Renegade no App2Car.



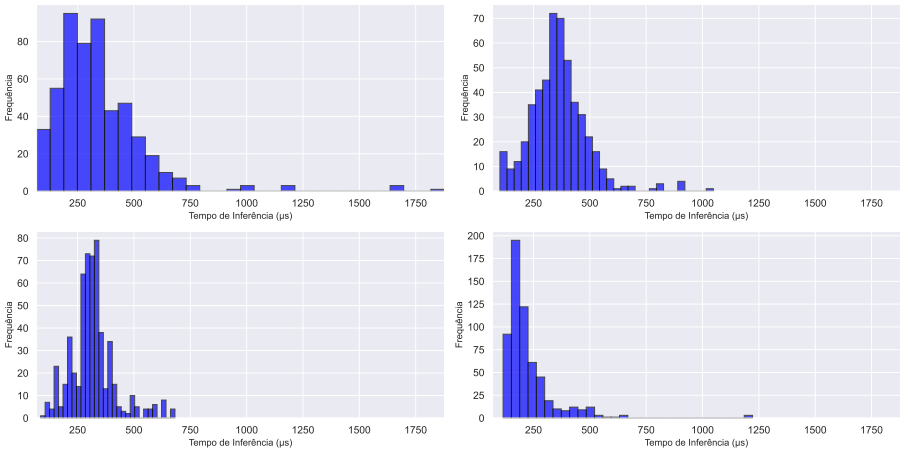
Fonte: Elaborado pelo Autor, 2025.

Figura B.2: Distribuição dos tempos de execução do Fiat Fastback no App2Car.



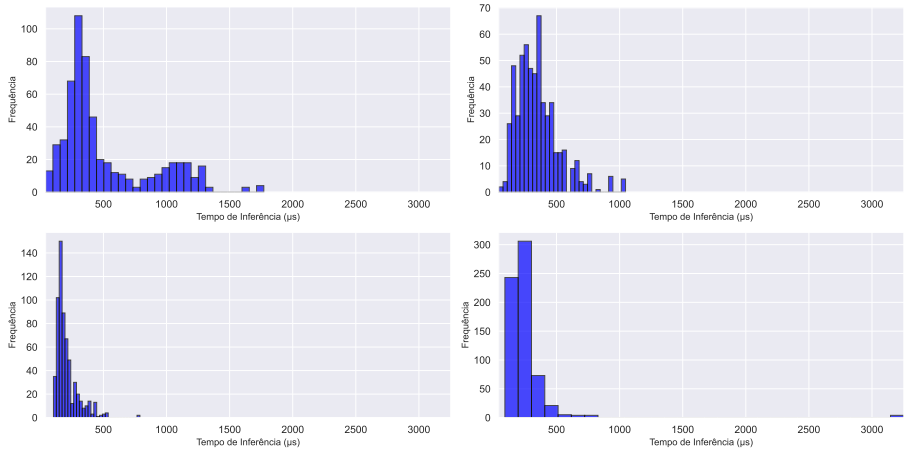
Fonte: Elaborado pelo Autor, 2025.

Figura B.3: Distribuição dos tempos de execução do Volkswagen Polo (Branco) no App2Car.



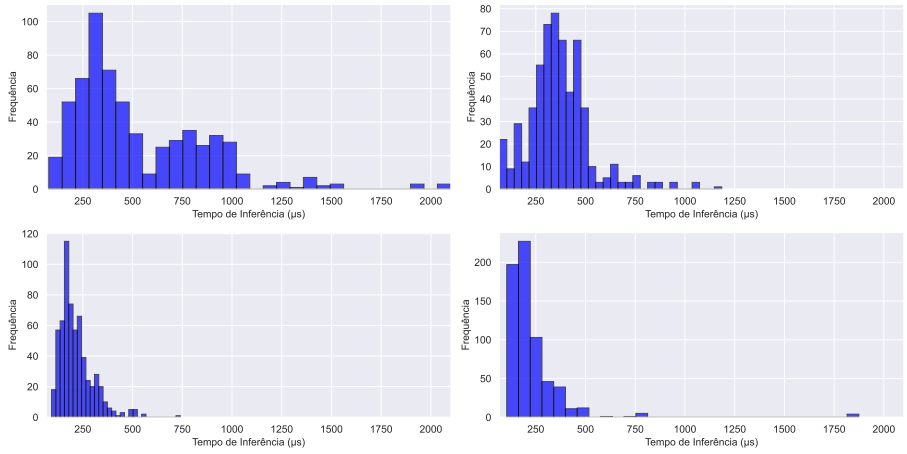
Fonte: Elaborado pelo Autor, 2025.

Figura B.4: Distribuição dos tempos de execução do Volkswagen Polo (Prata) no App2Car.



Fonte: Elaborado pelo Autor, 2025.

Figura B.5: Distribuição dos tempos de execução do Toyota Hilux no App2Car.



Fonte: Elaborado pelo Autor, 2025.