



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programa de Pós-Graduação em Informática

Isabela Regina Aguilar

**BASES DE DADOS COM CENÁRIOS HETEROGÊNEOS
PARA ANÁLISE E AVALIAÇÃO DE INTRUSÃO EM
CONTROLLER AREA NETWORKS DE VEÍCULOS
AUTÔNOMOS**

Belo Horizonte

2026

Isabela Regina Aguilar

**BASES DE DADOS COM CENÁRIOS HETEROGÊNEOS
PARA ANÁLISE E AVALIAÇÃO DE INTRUSÃO EM
CONTROLLER AREA NETWORKS DE VEÍCULOS
AUTÔNOMOS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Henrique Cota de Freitas

Belo Horizonte

2026

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

A283b Aguilar, Isabela Regina
Bases de dados com cenários heterogêneos para análise e avaliação de
intrusão em controller area networks de veículos autônomos / Isabela Regina
Aguilar. Belo Horizonte, 2026.
93 f. : il.

Orientador: Henrique Cota de Freitas
Dissertação (Mestrado) - Pontifícia Universidade Católica de Minas Gerais.
Programa de Pós-Graduação em Informática

1. Tecnologia da informação e da comunicação. 2. Veículos autônomos -
Sistema de informação. 3. Segurança de sistemas. 4. Redes de computadores -
Medidas de segurança. 5. Aprendizado do computador. 6. Banco de dados. I.
Freitas, Henrique Cota de. II. Pontifícia Universidade Católica de Minas Gerais.
Programa de Pós-Graduação em Informática. III. Título.

SIB PUC MINAS

CDU: 681.3.065

Ficha catalográfica elaborada por Gabriela Fernandes dos Santos Dias - CRB 6/4473

Isabela Regina Aguilar

**BASES DE DADOS COM CENÁRIOS HETEROGÊNEOS
PARA ANÁLISE E AVALIAÇÃO DE INTRUSÃO EM
CONTROLLER AREA NETWORKS DE VEÍCULOS
AUTÔNOMOS**

Dissertação apresentada ao Programa de
Pós-Graduação em Informática da Pontifí-
cia Universidade Católica de Minas Gerais,
como requisito parcial para obtenção do tí-
tulo de Mestre em Informática.

Prof. Dr. Henrique Cota de Freitas
PUC Minas (Orientador)

Prof.^a Dr.^a Cristiane Neri Nobre
PUC Minas

Prof. Dr. Adenauer Correa Yamin
Universidade Federal de Pelotas

Prof. Dr. Marcos Tulio Amaris González
Universidade Federal do Pará

Belo Horizonte, 02 de Junho de 2026.

Dedico este trabalho a Deus, sem Ele nada disso seria possível. À minha família, por todo amor, cuidado e incentivo. Essa conquista também é de vocês.

AGRADECIMENTOS

Começo este agradecimento expressando minha profunda gratidão a Deus, por ter me sustentado até aqui. Esta dissertação é a realização de um grande sonho e tenho a certeza de que é só o começo.

Aos meus pais, Gilmar e Zilda, e à minha irmã Carolinne, muito obrigada por todo apoio nessa caminhada e por estarem comigo em todos os momentos. Vocês são um presente de Deus na minha vida e sou muito grata a Ele por fazer parte dessa família. Amo vocês.

Aos meus familiares e amigos, agradeço por todo incentivo e carinho. Vocês foram fundamentais para que eu chegasse até aqui.

À PUC Minas pelo comprometimento e pelo apoio em diversos momentos da minha trajetória acadêmica. Agradeço ao Programa de Pós-graduação em Informática (PP-GINF), sua coordenação, secretaria, professores e demais colaboradores.

Ao meu orientador, Prof. Dr. Henrique Cota de Freitas, que me auxiliou em todos os momentos e sempre acreditou no meu potencial.

Às agências de fomento, agradeço o apoio financeiro essencial para a realização desta pesquisa: à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), código de financiamento 001; ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo 402837/2024-0; e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), processo APQ-05058-23.

Agradeço a todos os colegas com quem tive o privilégio de trabalhar e desejo sucesso a todos vocês.

Por fim, tenho a certeza que essa conquista não é só minha. Deus realiza sonhos e Ele faz infinitamente mais do que pedimos ou pensamos. Sei que tudo o que eu vivi e que viverei nunca foi e nunca será sobre mim e sim sobre Ele.

“Grandes coisas fez o SENHOR por nós, e, por isso, estamos alegres.” Salmos 126:3

“The problem is not the technology. The problem is the person or people who use it.”

Billy Graham

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

De acordo com a Portaria Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) nº 2.664, de 6 de março de 2026 (CNPq, 2026), com o objetivo de estabelecer a Política de Integridade na Atividade Científica do CNPq, declara-se que foram utilizadas ferramentas de Inteligência Artificial Generativa (IAG) em etapas específicas da metodologia deste trabalho, conforme detalhado no Capítulo 4 e para auxiliar eventuais traduções de textos em língua inglesa.

É válido destacar que todo o conteúdo gerado com auxílio de IAG foi revisado, validado e é de inteira responsabilidade da autora.

RESUMO

Sistemas eletrônicos estão cada vez mais integrados aos veículos modernos. Essa integração tem sido acelerada pelo aumento no desenvolvimento de veículos autônomos e conectados. Como consequência disso, esses sistemas ficam expostos a uma superfície de ataque maior. Nos dias atuais, as redes veiculares passaram a se comunicar inclusive com infraestruturas externas, tornando a segurança cibernética automotiva um requisito crítico para a mobilidade. Nesse contexto, a rede *Controller Area Network* (CAN), baseada no protocolo de mesmo nome, adotado pela indústria automotiva há décadas, é considerada como um dos maiores pontos de vulnerabilidade nos veículos. Sua concepção original não contempla mecanismos de autenticação ou controle de acesso, expondo a rede veicular interna a diversos ataques. Para detectar e responder a essas ameaças, abordagens baseadas em aprendizado de máquina têm se destacado pela capacidade de identificar padrões anômalos. No entanto, a eficácia dessas soluções depende diretamente da representatividade dos conjuntos de dados utilizados para treinamento desses modelos. As bases de dados disponíveis, em sua maioria, são coletadas em condições homogêneas, seja utilizando um único veículo, ou em ambiente controlado. Essa limitação compromete a construção e treinamento de modelos para obterem a habilidade de se adaptarem aos diversos cenários operacionais que, principalmente, os veículos autônomos enfrentam. Diante dessa lacuna, este trabalho propõe uma metodologia para a criação de bases de dados mais representativas que podem ser aplicadas na detecção de intrusões em redes CAN. Para isso, foi utilizado o simulador *Car Learning to Act* (CARLA) com o objetivo de produzir 45 cenários com diferentes ambientes, condições climáticas e perfis de ataque. A base resultante reúne mais de 3 milhões de instâncias. Além disso, quatro algoritmos foram avaliados, sendo eles: *Decision Tree*, *Random Forest*, *XGBoost* e *Convolutional Neural Network* (CNN). Entre eles, o que apresentou melhor resultado foi a CNN, com *F1-score* médio de 0,87. Os resultados obtidos neste trabalho evidenciam que bases com cenários heterogêneos podem contribuir com o desenvolvimento de mecanismos de detecção eficazes para veículos autônomos.

Palavras-chave: Veículos autônomos. Segurança cibernética. *Controller Area Network*. Detecção de intrusão. Aprendizado de máquina. Bases de dados heterogêneas.

ABSTRACT

Electronic systems are increasingly integrated into modern vehicles. This integration has been accelerated by the growing development of autonomous and connected vehicles. As a consequence, these systems are exposed to a larger attack surface. Nowadays, vehicular networks have come to communicate with external infrastructures as well, making automotive cybersecurity a critical requirement for mobility. In this context, the *Controller Area Network* (CAN), based on the protocol of the same name, adopted by the automotive industry for decades, is considered one of the most significant vulnerability points in vehicles. Its original design does not include authentication mechanisms or access control, exposing the internal vehicular network to various attacks. To detect and respond to these threats, machine learning-based approaches have stood out for their ability to identify anomalous patterns. However, the effectiveness of these solutions depends directly on the representativeness of the datasets used to train these models. Most available datasets are collected under homogeneous conditions, either using a single vehicle or in a controlled environment. This limitation hinders the development and training of models capable of adapting to the diverse operational scenarios faced, in particular, by autonomous vehicles. To address this gap, this work proposes a methodology for creating more representative datasets that can be applied to intrusion detection in CAN networks. To this end, the *Car Learning to Act* (CARLA) simulator was used to produce 45 scenarios with different environments, weather conditions, and attack profiles. The resulting dataset comprises more than 3 million instances. Furthermore, four algorithms were evaluated: *Decision Tree*, *Random Forest*, *XGBoost*, and *Convolutional Neural Network* (CNN). Among them, the CNN achieved the best results, with an average *F1-score* of 0.87. The findings of this work demonstrate that datasets with heterogeneous scenarios can contribute to the development of effective intrusion detection mechanisms for autonomous vehicles.

Keywords: Autonomous vehicles. Cybersecurity. *Controller Area Network*. Intrusion detection. Machine learning. Heterogeneous datasets.

LISTA DE FIGURAS

FIGURA 1 – Este diagrama ilustra a conectividade de um veículo e suas vulnerabilidades. O sistema é dividido em três camadas principais: Conectividade Externa (Direita/Topo), Barramento CAN e Domínios (Centro), Acesso Físico via OBD-II (Esquerda). As setas azuis mostram o fluxo operacional seguro, enquanto as setas vermelhas expõem os caminhos de exploração que podem comprometer desde o entretenimento até a segurança física do condutor.	16
FIGURA 2 – Representação dos níveis de autonomia veicular segundo a classificação da SAE. Os níveis variam de 0 (sem automação) a 5 (autonomia total), indicando o grau de controle e intervenção no veículo. Adaptado de (FAWOLE; RAWAT, 2024)	23
FIGURA 3 – Estrutura do quadro de dados CAN. Adaptado de (LOKMAN; OTHMAN; ABU-BAKAR, 2019).	25
FIGURA 4 – Comparação entre as arquiteturas de <i>Intrusion Detection Systems</i> (IDSs) baseados em assinatura e baseados em anomalias. Adaptado de (BANGUI; GE; BUHNOVA, 2022).	32
FIGURA 5 – Etapas metodológicas para elaboração do conjunto de dados. O fluxo parte da simulação no CARLA e percorre a conversão para o formato CAN, a injeção controlada de ataques, o pré-processamento e balanceamento, até a avaliação comparativa dos algoritmos de classificação.	43
FIGURA 6 – Uma matriz 3x3 comparando as simulações em diferentes condições climáticas (Chuva/Neblina, Neve e Sol) em três cenários de simulação distintos: Urbano (Town03), Rodovia (Town04) e Rural (Town07).	56
FIGURA 7 – Distribuição de instâncias por categoria de ataque na base consolidada antes do balanceamento (5.399.832 registros). O <i>Denial of Service</i> (DoS) representa a maior parcela (2.116.707 registros, 39,2%) por causa da injeção massiva de mensagens que caracteriza esse ataque. O tráfego Normal corresponde a apenas 19,0% do total, refletindo o desequilíbrio entre operação legítima e maliciosa nos cenários de teste.	65

FIGURA 8 – Contagem de instâncias por combinação de ambiente e condição climática, mostrando a cobertura equilibrada dos 45 cenários. A variação entre a célula de menor volume (Urbano & Seco, 580.891 registros) e a de maior volume (Rodovia & Neve, 650.115 registros) é de apenas 12%.	65
FIGURA 9 – <i>F1-score</i> médio por algoritmo e categoria de ataque, consolidando todos os 45 cenários. Cada eixo representa uma categoria e a área preenchida reflete o desempenho geral de cada modelo. A CNN apresenta a maior cobertura em todas as dimensões, enquanto a <i>Decision Tree</i> apresenta as maiores quedas nas categorias de <i>Spoofing</i> , onde a relação entre sinais é essencial para a detecção.	67
FIGURA 10 – <i>F1-score</i> médio da CNN para cada combinação de ambiente e condição climática. O cenário mais favorável é Rodovia & Seco (0,90) e o mais difícil é Urbano & Neve (0,81).	76
FIGURA 11 – Comparação do <i>F1-score</i> da CNN entre o <i>dataset</i> proposto e o <i>Car-Hacking Dataset</i> (SEO; SONG; KIM, 2018) por categoria de ataque. O <i>dataset</i> proposto supera o <i>Car-Hacking</i> nas categorias de <i>Spoofing</i> (RPM e Gear), onde a variedade de cenários expõe o modelo a variações legítimas ausentes na base de referência.	81

LISTA DE TABELAS

TABELA 1 – Comparação entre bases de dados para detecção de intrusão em redes CAN. A proposta destaca-se pela operação autônoma, variação sistemática de via e clima, e por incluir metadados contextuais (15 atributos).	38
TABELA 2 – Comparação entre abordagens metodológicas para análise e avaliação de <i>Intrusion Detection Systems</i> (IDSs) em redes CAN, destacando algoritmos utilizados, métricas reportadas e protocolo de validação. A proposta deste trabalho diferencia-se pela avaliação comparativa de quatro algoritmos com otimização automática via Optuna e validação cruzada em 45 cenários.	41
TABELA 3 – Limites de velocidade mínima e máxima adotados para cada combinação de ambiente e condição climática nos cenários simulados, definidos com base no Código de Trânsito Brasileiro (CTB) e na literatura sobre controle autônomo em condições adversas. Reduções de velocidade em chuva e neve refletem o comportamento esperado de um sistema autônomo adaptativo. .	47
TABELA 4 – Distribuição de registros por categoria antes do balanceamento no conjunto de treinamento bruto. Observa-se a predominância de registros de ataque (aproximadamente 81%) em relação aos normais (aproximadamente 19%). Esse desequilíbrio motivou a aplicação de <i>undersampling</i> combinado com Synthetic Minority Over-sampling Technique (SMOTE).	49
TABELA 5 – Configuração do ambiente computacional e ferramentas utilizadas . . .	53
TABELA 6 – Hiperparâmetros finais selecionados pelo Optuna para cada algoritmo, obtidos por busca bayesiana com <i>Tree-structured Parzen Estimator</i> (TPE). .	54
TABELA 7 – Descrição das colunas presentes no arquivo <code>combined_can_dataset.csv</code> , com tipo de dado e valores possíveis para cada campo. As colunas <code>road_type</code> e <code>climate</code> são adicionadas pelo <i>script</i> de mesclagem; <code>flag</code> e <code>attack_type</code> operam em níveis distintos de rotulagem, conforme detalhado na Seção 5. .	60
TABELA 8 – Contagem de registros por combinação de ambiente e condição climática, antes do balanceamento.	61

TABELA 9 – Distribuição de registros por categoria de ataque antes do balanceamento (5.399.832 registros). O DoS concentra o maior volume em razão da injeção massiva de mensagens que caracteriza esse vetor, enquanto <i>RPM Spoofing</i> e <i>Gear Spoofing</i> apresentam os menores suportes.	61
TABELA 10 – Divisão treino e teste antes e após o balanceamento. A separação foi feita por cenário completo para evitar <i>data leakage</i> . O balanceamento foi aplicado exclusivamente ao conjunto de treinamento, preservando a distribuição original no conjunto de teste.	62
TABELA 11 – Distribuição de registros por categoria após o balanceamento do conjunto de treinamento. A combinação de <i>undersampling</i> e <i>SMOTE</i> igualou as cinco categorias em 511.225 instâncias cada, totalizando 2.556.125 registros.	62
TABELA 12 – <i>F1-score</i> médio $\pm$ desvio padrão por modelo e categoria de ataque, calculado sobre as avaliações individuais dos 45 cenários. O desvio padrão reflete a variação de desempenho entre cenários: classes de difícil detecção (<i>Spoofing</i>) e modelos mais simples apresentam maior dispersão. Os valores elevados de <i>Denial of Service</i> (DoS) ($\geq 0,98$) refletem avaliações isoladas; na avaliação conjunta do <i>dataset</i> consolidado (Tabela 25), os valores são menores em razão da competição simultânea entre todas as classes. A CNN obtém a maior média global (0,87) com a menor dispersão ($\pm 0,02$).	66
TABELA 13 – Precisão e <i>recall</i> detalhados para o melhor cenário combinado (Rodovia & Seco, <i>F1-score</i> médio CNN = 0,90) e o pior (Urbano & Neve, <i>F1-score</i> médio CNN = 0,81), evidenciando a amplitude real de variação do sistema quando ambiente e condição climática se combinam.	68
TABELA 14 – <i>F1-score</i> médio $\pm$ desvio padrão por modelo e condição climática. A neve apresenta maior dispersão que o clima seco, refletindo a interação variável entre a dinâmica de patinagem das rodas e os diferentes perfis de ataque em cada ambiente.	70
TABELA 15 – Resultados detalhados por categoria de ataque em condição climática seca, que serve como referência de operação ideal. A CNN alcança <i>F1-score</i> médio de 0,89; a <i>Decision Tree</i> estabelece a linha de base em 0,72.	70
TABELA 16 – Resultados detalhados por categoria de ataque em condição de chuva	

com neblina. Degradação de 2 a 3 pontos percentuais em relação ao clima seco para todos os modelos, com a CNN mantendo desempenho de 0,87. . .	71
TABELA 17 – Resultados detalhados por categoria de ataque em condição de neve. Cenário mais desafiador: queda de 7 a 9 pontos percentuais em relação ao seco, com aumento de falsos positivos em <i>RPM Spoofing</i> causado pelo <i>spin-up</i> legítimo das rodas em pista com baixa aderência.	72
TABELA 18 – <i>F1-score</i> médio $\pm$ desvio padrão por modelo e ambiente, calculado sobre os 15 cenários de cada grupo viário (3 condições climáticas $\times$ 5 perfis de ataque). O ambiente urbano apresenta maior dispersão em razão da alta variabilidade das manobras, que interage de forma diferente com cada perfil de ataque e condição climática.	73
TABELA 19 – Resultados detalhados por categoria de ataque em rodovia. A estabilidade dos sinais em rodovias favorece todos os algoritmos: CNN (0,90) e <i>XGBoost</i> (0,87) apresentam os melhores resultados absolutos.	73
TABELA 20 – Resultados detalhados por categoria de ataque em estrada rural. Desempenho intermediário entre rodovia e urbano, com CNN em 0,88 e <i>XGBoost</i> em 0,85, refletindo a variabilidade moderada das manobras nesse ambiente.	74
TABELA 21 – Resultados detalhados por categoria de ataque em ambiente urbano. A <i>Decision Tree</i> sofre a degradação mais severa (0,63), evidenciando a insuficiência de modelos simples frente à alta sobreposição de classes nesse contexto.	75
TABELA 22 – <i>F1-score</i> médio da CNN para cada combinação de ambiente e condição climática. Dois gradientes simultâneos: queda de rodovia para urbano e de seco para neve. Amplitude total de 9 pontos percentuais entre o melhor (Rodovia & Seco, 0,90) e o pior cenário (Urbano & Neve, 0,81).	75
TABELA 23 – <i>F1-score</i> por categoria de ataque para cada combinação de ambiente e condição climática, complementando as Tabelas 14 e 18 com a perspectiva cruzada. Dois gradientes simultâneos são observados: queda de Rodovia para Urbano e de Seco para Neve. A combinação Urbano & Neve representa o pior caso, com <i>Gear Spoofing</i> e <i>RPM Spoofing</i> como as categorias mais afetadas em todos os algoritmos. Valores em negrito referem-se à CNN. . .	77
TABELA 24 – <i>F1-score</i> da CNN por categoria de ataque e combinação de ambi-	

ente e condição climática. Detalha como cada categoria contribui para os gradientes observados na Tabela 22. <i>RPM Spoofing</i> e <i>Gear Spoofing</i> apresentam as maiores quedas acumuladas (até 15 pp entre Rodovia &Seco e Urbano &Neve), enquanto DoS mantém a maior resiliência em todos os cenários.	78
TABELA 25 – Relatório de classificação completo para o <i>dataset</i> consolidado (45 cenários, 639.031 registros do subconjunto de avaliação final, correspondente aos 1.079.966 registros de teste originais após reserva do conjunto de validação utilizado pelo Optuna). Apresenta precisão, <i>recall</i> , <i>F1-score</i> e suporte por categoria para os quatro algoritmos avaliados.	79
TABELA 26 – Desempenho dos quatro algoritmos no <i>Car-Hacking Dataset</i> (SEO; SONG; KIM, 2018), sob protocolo experimental equivalente ao adotado para o <i>dataset</i> proposto.	80

LISTA DE ABREVIATURAS E SIGLAS

ACK - *Acknowledgement*

ADAS - *Advanced Driver Assistance Systems*

CAN - *Controller Area Network*

CAN-FD - *CAN with Flexible Data-Rate*

CANH - *CAN High*

CANL - *CAN Low*

CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CARLA - *Car Learning to Act*

CAV - *Connected and Autonomous Vehicle*

CNN - *Convolutional Neural Network*

CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico

CRC - *Cyclic Redundancy Check*

CTB - Código de Trânsito Brasileiro

DBC - *Database CAN*

DLC - *Data Length Code*

DNN - *Deep Neural Network*

DoS - *Denial of Service*

ECU - *Electronic Control Unit*

EOF - *End of Frame*

FAPEMIG - Fundação de Amparo à Pesquisa do Estado de Minas Gerais

GPU - *Graphics Processing Unit*

HCRL - *Hacking and Countermeasure Research Lab*

IAG - Inteligência Artificial Generativa

IDS - *Intrusion Detection System*

LIN - *Local Interconnect Network*

LSTM - *Long Short-Term Memory*

OBD-II - *On-Board Diagnostics II*

OSI - *Open Systems Interconnection*

OTIDS - *On-board Time-based Intrusion Detection System*

RPM - *Revolutions Per Minute*

SAE - *Society of Automotive Engineers*

SMOTE - *Synthetic Minority Over-sampling Technique*

SOF - *Start of Frame*

TPE - *Tree-structured Parzen Estimator*

V2I - *Vehicle-to-Infrastructure*

V2V - *Vehicle-to-Vehicle*

V2X - *Vehicle-to-Everything*

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Problema de pesquisa	17
1.2	Hipóteses	17
1.3	Objetivos	18
1.3.1	<i>Objetivo geral.....</i>	18
1.3.2	<i>Objetivos específicos</i>	18
1.4	Justificativa	19
1.5	Contribuições da dissertação.....	19
1.6	Organização da dissertação	20
2	REFERENCIAL TEÓRICO.....	22
2.1	Segurança cibernética em veículos autônomos	22
2.2	Controller Area Network	24
2.2.1	<i>Características e funcionamento</i>	24
2.2.2	<i>Outros protocolos de comunicação veicular</i>	26
2.2.3	<i>Arquitetura de redes veiculares e vetores de ataque ao barramento CAN</i>	27
2.2.4	<i>Interfaces externas e pontos de entrada</i>	28
2.2.5	<i>Vulnerabilidades do barramento CAN</i>	29
2.2.6	<i>Comunicação CAN em veículos autônomos</i>	30
2.2.7	<i>Como a rede CAN é comprometida na prática</i>	30
2.3	Sistemas de detecção de intrusão	31
2.3.1	<i>Tipos de sistemas de detecção de intrusão</i>	31
2.3.2	<i>Desafios na detecção de intrusão em redes CAN</i>	32
2.4	O papel das bases de dados no desenvolvimento de IDSs	34
2.4.1	<i>Heterogeneidade de cenários em segurança veicular</i>	34
2.4.2	<i>Desafios na geração de bases representativas</i>	34
3	TRABALHOS RELACIONADOS.....	36

3.1	<i>Datasets</i> para detecção de intrusão veicular	36
3.1.1	<i>Waymo Open Dataset</i>	39
3.2	Abordagens de detecção e classificação de intrusões	39
3.3	Lacunas identificadas e posicionamento desta pesquisa	40
4	MATERIAIS E MÉTODOS	43
4.1	Utilização de Inteligência Artificial	44
4.2	Construção e organização das bases de dados	44
4.2.1	<i>Simulação de cenários com CARLA</i>	45
4.2.2	<i>Caracterização dos cenários climáticos e parâmetros de velocidade</i>	46
4.2.3	<i>Transformação em frames CAN com CANTools</i>	47
4.2.4	<i>Injeção e modelagem dos cenários de ataque</i>	48
4.3	Pré-processamento dos dados	48
4.3.1	<i>Limpeza e codificação dos dados</i>	50
4.4	Algoritmos de classificação para detecção de intrusões	50
4.5	Métricas e protocolos de avaliação	51
4.6	Ambiente e configuração dos experimentos	53
5	DESENVOLVIMENTO E CARACTERIZAÇÃO DO DATASET DE INTRUSÃO CAN	55
5.1	Configuração do simulador CARLA e ambiente experimental	55
5.2	Estratégia de construção de cenários	57
5.3	Conversão de mensagens para o padrão CAN 2.0B	57
5.4	Influência do clima nos dados	57
5.5	Implementação técnica dos ataques cibernéticos	58
5.6	Unificação, rotulagem e estrutura final do <i>dataset</i>	59
5.6.1	<i>Estrutura das colunas</i>	60
5.6.2	<i>Semântica dos campos flag e attack_type</i>	60
5.6.3	<i>Caracterização estatística e preparação final do dataset</i>	61
6	RESULTADOS E DISCUSSÃO	64
6.1	Análise descritiva do <i>dataset</i>	64
6.2	Desempenho dos algoritmos de classificação	66

6.2.1	<i>Comparação geral de desempenho</i>	66
6.2.2	<i>Análise qualitativa dos padrões de erro</i>	69
6.2.3	<i>Resultados em diferentes condições climáticas</i>	69
6.2.4	<i>Resultados por ambiente</i>	72
6.2.5	<i>Efeito combinado de via e clima sobre a CNN</i>	75
6.2.6	<i>Resultados consolidados do dataset combinado</i>	78
6.3	<i>Comparação com o Car-Hacking Dataset</i>	79
6.3.1	<i>Desempenho no Car-Hacking Dataset</i>	79
6.3.2	<i>Comparação com o dataset proposto</i>	80
6.4	<i>Interpretação técnica e limitações observadas</i>	81
6.4.1	<i>Por qual motivo a CNN se destaca</i>	81
6.4.2	<i>Dificuldade na detecção de Gear Spoofing em ambientes urbanos</i>	82
6.4.3	<i>Influência climática e necessidade de contexto</i>	82
6.4.4	<i>Eficiência do XGBoost para ataques Fuzzing</i>	83
6.4.5	<i>Detectar versus interromper ataques DoS</i>	83
6.5	<i>Impacto da diversidade dos cenários no desempenho dos modelos</i>	83
6.6	<i>Limitações dos experimentos e caminhos futuros</i>	84
7	CONCLUSÕES	85
7.1	<i>Trabalhos futuros</i>	87
7.1.1	<i>Produção decorrente da dissertação</i>	88
	REFERÊNCIAS	89

1 INTRODUÇÃO

O setor automotivo vem enfrentando uma grande transformação impulsionada pelos avanços tecnológicos das últimas décadas. O mercado global de veículos autônomos, avaliado em USD 89,83 bilhões em 2025, deverá atingir USD 719,35 bilhões até 2033, com uma taxa de crescimento anual de 29,7% (SkyQuestt, 2025). Esses números comprovam o aumento da relevância e adoção dessas tecnologias em escala mundial. Com o surgimento dos *Connected and Autonomous Vehicles* (CAVs), tecnologias como inteligência artificial, redes de sensores e sistemas integrados desempenham um importante papel na busca por maior segurança, eficiência e conveniência para os usuários.

De acordo com Liu et al. (2017), os CAVs são capazes de realizar tarefas complexas, como navegação autônoma e controle dinâmico, com mínima ou nenhuma intervenção humana. Entretanto, essa autonomia e conectividade também trazem novos desafios, principalmente relacionados à segurança das redes internas do veículo.

A rede *Controller Area Network* (CAN) é uma arquitetura de comunicação criada na década de 1980 para interligar componentes eletrônicos. Ela é adotada para permitir a comunicação interna entre *Electronic Control Units* (ECUs), sensores e atuadores, sendo responsável por trafegar desde dados de motores e transmissão até sinais de freios e direção do veículo. Tornou-se popular, pois combina baixo custo de implementação e confiabilidade operacional, fatores que fazem com que permaneça como padrão dominante nos veículos modernos, mesmo diante do avanço das ameaças digitais e da dificuldade de substituição em larga escala (BUSCEMI et al., 2023). Contudo, essa ampla adoção faz com que a rede CAN seja considerada uma das principais portas de entrada para vulnerabilidades nos veículos autônomos modernos.

Embora a rede CAN opere internamente, ela não está completamente isolada do ambiente externo. Existem interfaces como o conector *On-Board Diagnostics II* (OBD-II), módulos de telemetria celular, sistemas de infoentretenimento com suporte a Bluetooth e Wi-Fi, e unidades de comunicação *Vehicle-to-Everything* (V2X) que, conforme descrito por Miller e Valasek (2014), Anwar et al. (2023), criam pontos de contato entre a rede interna e redes externas, ampliando a superfície de ataque disponível a agentes maliciosos. Essa interconexão, ilustrada na Figura 1*, permite que um potencial invasor alcance a rede CAN por caminhos que vão além do acesso físico direto ao veículo.

*Veículo: <<https://pt.vecteezy.com/arte-vetorial/36254498-vista-de-cima-do-carro>>
 Invasor: <<https://br.pinterest.com/pin/659988520382697115/>>
 Nuvem: <https://www.magnific.com/br/icone/nuvem-wifi_72981>
 Antena: <https://www.flaticon.com/br/icone-gratis/antena-parabolica_1082877>
 Bluetooth, Wi-Fi: <<https://www.canva.com/>>
 OBD-II: <https://commons.wikimedia.org/wiki/File:OBD-II_type_B_female_connector_shape.svg>

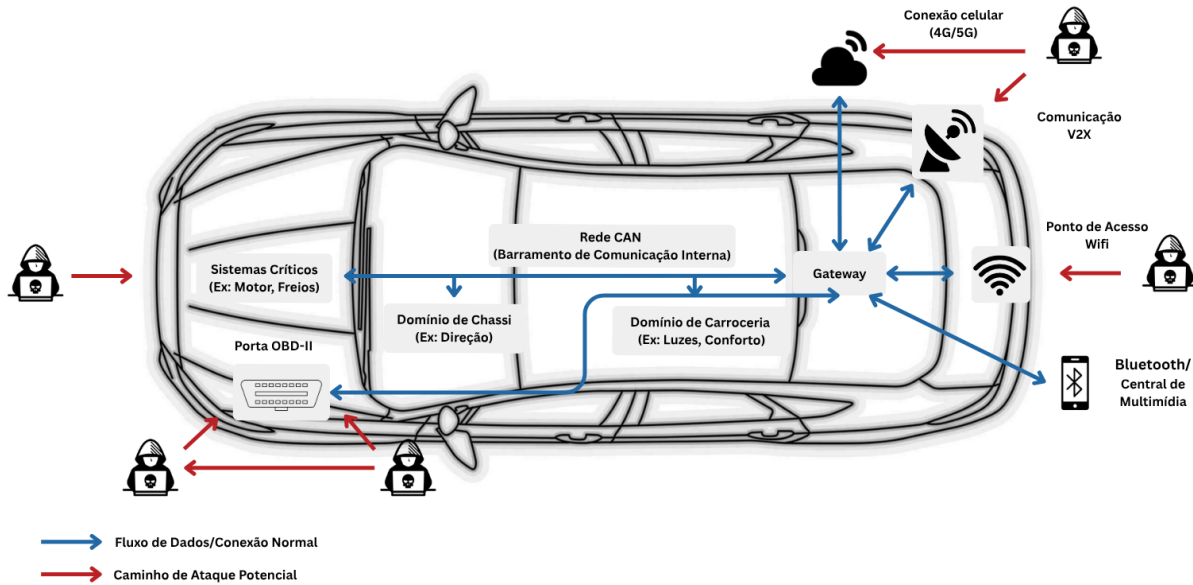


Figura 1 – Este diagrama ilustra a conectividade de um veículo e suas vulnerabilidades. O sistema é dividido em três camadas principais: Conectividade Externa (Direita/Topo), Barramento CAN e Domínios (Centro), Acesso Físico via OBD-II (Esquerda). As setas azuis mostram o fluxo operacional seguro, enquanto as setas vermelhas expõem os caminhos de exploração que podem comprometer desde o entretenimento até a segurança física do condutor.

Na prática, ataques à rede CAN frequentemente se iniciam com a exploração de uma interface externa menos protegida e escalam até sistemas críticos, como apresentado por Miller e Valasek (2015), em que obtiveram controle remoto de funções de um veículo em movimento ao explorar uma vulnerabilidade no módulo de conectividade. Uma vez que o acesso à rede é obtido, os ataques podem assumir diferentes formas, desde injeção de mensagens e negação de serviço até ataques mais sofisticados (CHO; SHIN, 2016), onde a detecção é ainda mais desafiadora.

Para mitigar esses riscos, os *Intrusion Detection Systems* (IDSs) são considerados uma solução com bons resultados. Esses sistemas monitoram o tráfego de dados em busca de padrões anômalos ou desvios que possam indicar atividades maliciosas (KORIUM et al., 2024). Contudo, a eficácia dos IDSs depende diretamente da qualidade, heterogeneidade e representatividade das bases de dados utilizadas no treinamento e na avaliação dos modelos de detecção.

Sendo assim, um dos grandes desafios no desenvolvimento de IDSs eficazes para veículos autônomos é a disponibilidade de bases de dados com cenários heterogêneos e abrangentes. Segundo Sanaboina, Naik e Rajiv (2023), é essencial que os *datasets* contenham uma variedade de cenários, incluindo diferentes tipos de ataques e condições operacionais diversas, a fim de preparar os sistemas para lidar com situações reais.

1.1 Problema de pesquisa

A modernização da conectividade em veículos modernos e o avanço de tecnologias como redes de sensores, inteligência artificial e sistemas integrados, proporcionam inúmeros benefícios aos seus usuários. No entanto, essa evolução também introduz vulnerabilidades críticas, especialmente relacionadas à segurança.

A rede CAN, amplamente utilizada para a comunicação interna entre componentes veiculares, representa um dos principais pontos de fragilidade devido à ausência de mecanismos nativos de autenticação e criptografia, representando um alto risco para a operação segura de veículos, principalmente os autônomos.

A necessidade de proteger as redes CAN de veículos autônomos contra ataques cibernéticos tem incentivado o desenvolvimento de IDSs. Esses sistemas dependem de algoritmos de aprendizado de máquina treinados com bases de dados representativas para identificar corretamente padrões anômalos e mitigar ameaças. Contudo, a escassez de *datasets* abrangentes e bem documentados reduz a eficácia e a aplicação desses sistemas em cenários reais.

Nesse contexto, a diversidade dos dados é um fator crítico, uma vez que a combinação de informações em diferentes cenários permite uma identificação mais completa e precisa tanto do comportamento normal quanto das anomalias em veículos.

Diante desse contexto, surgem as seguintes Questões de Pesquisa (QP):

- **QP 1:** Como desenvolver e estruturar bases de dados com cenários diversificados que representem adequadamente o comportamento normal e cenários de intrusão em redes CAN de veículos autônomos?
- **QP 2:** Quais cenários de operação de um veículo autônomo apresentam um desafio maior na identificação de padrões de intrusão?
- **QP 3:** De que forma a diversidade dos dados afeta a eficácia dos métodos de detecção de intrusão?

Estas questões são resultado de uma fragilidade relevante na literatura, que é a ausência de bases de dados com informações diversificadas que possam contribuir para o desenvolvimento de soluções de segurança mais eficazes e confiáveis para veículos autônomos.

1.2 Hipóteses

Este trabalho é fundamentado nas seguintes hipóteses:

- **Hipótese 1:** É possível desenvolver bases de dados com cenários heterogêneos a partir do barramento CAN que sejam mais informativas e representativas que as bases disponíveis até o desenvolvimento deste trabalho, permitindo melhor análise e avaliação de intrusões.
- **Hipótese 2:** As características dos dados coletados no barramento CAN, como diversidade de mensagens e presença de padrões anômalos, impactam diretamente na eficácia dos IDSs.
- **Hipótese 3:** As bases de dados com cenários heterogêneos permitem uma avaliação mais completa entre modelos de aprendizado de máquina para detecção de intrusões, ao expô-los a diferentes padrões e variações em redes CAN.

1.3 Objetivos

1.3.1 *Objetivo geral*

Este trabalho tem como objetivo principal desenvolver e avaliar uma base de dados com diversos cenários operacionais e de intrusão, baseada nas comunicações do barramento CAN de veículos autônomos. A partir dessa base de dados, busca-se realizar uma análise detalhada para compreender o quanto a representatividade dessa base de dados pode contribuir para a eficácia de algoritmos de aprendizado de máquina reconhecidos pelo seu amplo uso no desenvolvimento de IDSs.

1.3.2 *Objetivos específicos*

Os objetivos específicos deste estudo são:

- Caracterizar e analisar padrões de tráfego de dados nas redes CAN que permitam diferenciar operações normais de atividades maliciosas, destacando indicadores relevantes para a detecção de intrusões.
- Desenvolver uma base de dados com cenários heterogêneos [†] por meio da coleta, processamento e integração de dados relacionados ao barramento CAN de veículos autônomos, contemplando tanto comportamentos normais quanto diversos cenários de ataque.
- Avaliar a aplicabilidade e a eficácia da base de dados construída para o desenvolvimento de IDSs em veículos conectados.

[†]Entende-se como cenários heterogêneos conjuntos de dados que combinam múltiplas dimensões de variação operacional, por exemplo, ambiente de condução, condição climática e perfil dos dados trafegados, sejam eles maliciosos ou não.

1.4 Justificativa

A segurança cibernética automotiva é um tema extremamente relevante devido à integração tecnológica nos veículos modernos. A conexão entre sistemas eletrônicos e redes de comunicação amplia consideravelmente a superfície de ataque para agentes maliciosos. O barramento CAN, desenvolvido décadas atrás sem preparação para lidar com as ameaças cibernéticas recentes, constitui um ponto frágil que pode comprometer a segurança dos passageiros e de terceiros.

Na literatura científica atual existe uma deficiência de bases de dados que contemplem cenários, comportamentos e situações variadas nas redes CAN. Embora existam *datasets* disponíveis na literatura, a maioria deles é restrita a cenários específicos e, em muitos casos, não é adequada para aplicações em veículos autônomos.

Além disso, com o passar do tempo se tornam desatualizados em relação às técnicas de ataque mais recentes. Essa realidade foi documentada em estudos, conforme apresentado por Lampe e Meng (2024a), que analisaram e compararam os principais *datasets* públicos de detecção de intrusão automotiva, evidenciando suas capacidades, mas também suas limitações, o que dificulta o desenvolvimento de soluções de detecção de intrusão eficazes em diferentes contextos.

A utilização de bases de dados com cenários heterogêneos pode ser justificada pela complexidade operacional dos veículos autônomos, onde o comportamento normal e anômalo varia significativamente conforme as condições de operação. Representar essa diversidade nos dados é essencial para desenvolver e avaliar sistemas de detecção.

Este trabalho propõe preencher essa lacuna desenvolvendo uma metodologia para criação, estruturação e análise de bases de dados com cenários heterogêneos específica para redes CAN de veículos autônomos. Esta contribuição é fundamental para o avanço do conhecimento em segurança automotiva e para apoiar o desenvolvimento de IDSs mais eficazes.

1.5 Contribuições da dissertação

Esta dissertação apresenta como contribuição principal a concepção e a avaliação de uma base de dados diversificada, contendo cenários heterogêneos, para redes CAN de veículos autônomos. Essa base abrange diversos cenários operacionais com variação de ambientes de condução (urbano, rural e rodovias), condições climáticas (clima seco, neve, chuva e neblina) e tipos de ataque. O objetivo é superar as limitações dos *datasets* existentes, que geralmente se restringem a um número significativamente menor de condições específicas.

Do ponto de vista prático, a base e a metodologia desenvolvidas podem ser adotadas pela indústria automotiva e por pesquisadores que buscam desenvolver, treinar ou avaliar IDSs sem a necessidade de infraestrutura dedicada para coleta de dados em veículos reais, reduzindo custos e acelerando o ciclo de desenvolvimento de soluções de segurança.

O trabalho também estabelece uma metodologia estruturada para coleta, processamento e organização de dados do barramento CAN em formatos adequados tanto para análise manual quanto para processamento computacional.

A utilização do simulador CARLA (DELAVERI; KHANZADA; KWON, 2025) como ambiente de coleta dos dados atribui ao trabalho uma contribuição adicional, que é a reprodutibilidade. Ao contrário de bases coletadas em veículos reais, cujas condições de coleta dificilmente podem ser replicadas com exatidão, os cenários simulados podem ser reexecutados por qualquer pesquisador.

Por fim, esses conhecimentos contribuem diretamente para o aprimoramento dos algoritmos de detecção, aumentando sua precisão e reduzindo resultados incorretos.

1.6 Organização da dissertação

Este trabalho está estruturado de forma a conduzir o leitor da fundamentação teórica até a avaliação dos resultados, passando pela construção e análise da base de dados proposta.

No **Capítulo 2**, são apresentados os fundamentos teóricos relacionados aos veículos autônomos, redes de comunicação veiculares, segurança cibernética automotiva, IDSs e ao barramento CAN, construindo a base conceitual necessária para o desenvolvimento da pesquisa.

O **Capítulo 3** aborda os trabalhos relacionados, onde são analisados estudos sobre bases de dados para detecção de intrusão em redes CAN, destacando suas características, limitações e como a abordagem proposta nesta pesquisa pretende avançar o estado da arte.

No **Capítulo 4**, são detalhados os materiais e métodos utilizados, incluindo a metodologia para coleta e processamento de dados do barramento CAN, bem como as técnicas de análise aplicadas para caracterização e avaliação da base de dados.

O **Capítulo 5** descreve as etapas de construção do *dataset*, detalhando o processo de desenvolvimento da base de dados, a simulação dos cenários operacionais e dos diferentes tipos de ataque, além dos procedimentos adotados para garantir a representatividade dos dados gerados.

No **Capítulo 6**, são apresentados o método de avaliação da base de dados e os

resultados dos algoritmos de classificação utilizando a base de dados gerada por meio das simulações, com a finalidade de exemplificar os resultados obtidos após o treinamento com a base proposta.

Por fim, o **Capítulo 7** traz as considerações finais, sintetizando as contribuições da pesquisa e apontando possíveis direções para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Esta seção apresenta o referencial teórico que fundamenta a pesquisa, reunindo conceitos e definições relacionados à segurança cibernética em veículos autônomos. São discutidos os principais aspectos da evolução dos veículos autônomos, as características e vulnerabilidades da *Controller Area Network* (CAN). Além disso, são abordados os sistemas de detecção de intrusão, destacando seus tipos, desafios e contribuições para a proteção contra ataques cibernéticos. O objetivo é oferecer uma base de conhecimento que auxilie as análises e propostas desenvolvidas nas seções seguintes deste trabalho.

2.1 Segurança cibernética em veículos autônomos

O desenvolvimento de veículos autônomos é um marco no setor de transporte, possibilitando, conforme Silva et al. (2024), novas formas de mobilidade que prometem aumentar a segurança, a eficiência e a conveniência dos usuários. Equipados com sensores avançados e sistemas de inteligência artificial, esses veículos conseguem realizar tarefas complexas de navegação, monitoramento do ambiente e tomada de decisões em tempo real, reduzindo a necessidade de intervenção humana.

A classificação dos veículos autônomos é baseada em níveis de automação, que variam de 0 a 5, conforme a definição da *Society of Automotive Engineers* (SAE)*. A Figura 2 ilustra essa classificação.

Os seis níveis de automação descrevem uma progressão contínua de controle humano para controle do sistema. No nível 0, não há automação, ou seja, o motorista assume controle total de todas as tarefas de condução, embora o veículo possa oferecer alertas informativos, como avisos de colisão iminente, sem atuar diretamente nos controles. No nível 1, o sistema auxilia o motorista em uma única tarefa de cada vez, como manutenção de faixa ou controle de velocidade adaptativo, mas o condutor permanece responsável pelas demais funções. No nível 2, o veículo pode executar simultaneamente duas ou mais tarefas de controle, como acelerar, frear e manter a centralização na faixa, mas o motorista deve manter atenção contínua e estar pronto para assumir o controle a qualquer momento.

No nível 3, o sistema monitora o ambiente de condução e gerencia todas as tarefas dinâmicas em determinadas condições, mas pode solicitar que o motorista retome o

*<https://www.sae.org/standards/content/j3016_202104/>

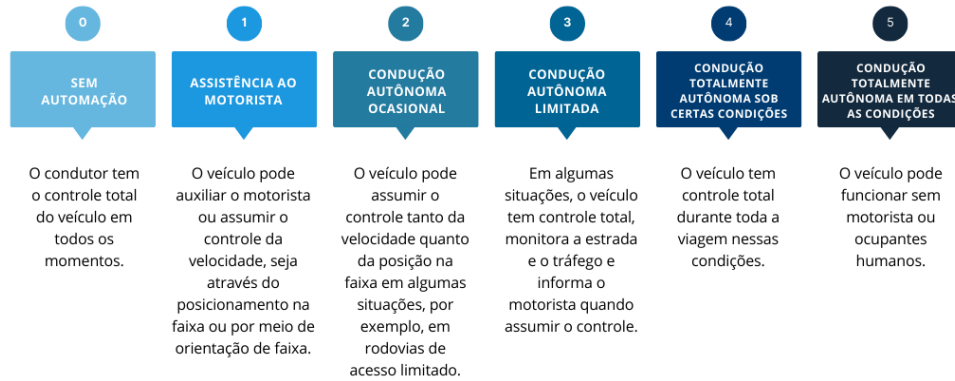


Figura 2 – Representação dos níveis de autonomia veicular segundo a classificação da SAE. Os níveis variam de 0 (sem automação) a 5 (autonomia total), indicando o grau de controle e intervenção no veículo. Adaptado de (FAWOLE; RAWAT, 2024)

controle quando necessário, ou seja, o condutor não precisa monitorar o ambiente continuamente, mas deve estar disponível para intervir. Por sua vez, o nível 4 representa automação de alta capacidade, em que o veículo opera de forma totalmente autônoma em domínios ou condições operacionais definidos, como zonas urbanas ou rodovias mapeadas, sem necessidade de intervenção humana nesse contexto. Por fim, o nível 5 representa a automação completa e irrestrita, em que o veículo é capaz de operar em todas as condições e cenários que um motorista seria capaz de enfrentar, sem qualquer intervenção humana.

Essa evolução tem consequências diretas para a segurança cibernética, pois à medida que o nível de automação aumenta, o volume de comunicação entre ECUs, sensores e sistemas de controle cresce significativamente, ampliando a superfície de ataque desses veículos. Ao mesmo tempo, a conectividade com dispositivos externos, redes *Vehicle-to-Vehicle* (V2V) e infraestruturas *Vehicle-to-Infrastructure* (V2I) expõe o veículo a vetores de ataque adicionais.

A complexidade dos veículos autônomos tem como resultado uma produção expressiva de dados vindos de diversos sensores, sistemas de controle e unidades de processamento. Essa diversidade, combinada com a variabilidade de cenários operacionais como condições urbanas e climáticas, cria grandes desafios para a segurança, em particular para a detecção de intrusões.

A cibersegurança em veículos autônomos é um aspecto crítico, pois, conforme Korium et al. (2024), falhas nos sistemas de controle podem resultar em consequências graves, como perda de direção ou manipulação indevida do veículo. Segundo Miller e Valasek (2014), os sistemas internos dos veículos autônomos incluem múltiplos pontos de conexão, desde ECUs até interfaces de comunicação como *Wi-Fi*, *Bluetooth* e redes de longa distância, todos vulneráveis a ataques cibernéticos.

O *Wi-Fi*, por exemplo, pode ser explorado para realizar ataques do tipo *man-in-the-middle*, interceptando ou modificando dados trocados entre o veículo e infraestruturas externas. O *Bluetooth*, por sua vez, é vulnerável a ataques de emparelhamento não autorizado, permitindo que um adversário fisicamente próximo envie comandos maliciosos a sistemas internos do veículo. Já as redes de longa distância, como as conexões celulares utilizadas para atualizações remotas, podem ser alvo de ataques de injeção de *firmware*, comprometendo o *software* de ECUs críticas (ANWAR et al., 2023).

Além de compreender os desafios de cibersegurança enfrentados pelos veículos autônomos, é fundamental analisar a infraestrutura de comunicação interna desses sistemas. A rede CAN representa o componente central dessa infraestrutura, constituindo, portanto, um elemento crítico tanto para a operação veicular quanto para potenciais vulnerabilidades de segurança.

2.2 Controller Area Network

A rede CAN, desenvolvida pela Bosch na década de 1980 e padronizada pela ISO 11898, conforme detalhado por Lampe e Meng (2023), é uma arquitetura de comunicação amplamente utilizada na comunicação intra-veicular, permitindo a troca de dados entre componentes como ECUs, sensores e atuadores por meio de um barramento compartilhado. Seu baixo custo de implementação e confiabilidade foram fatores determinantes para sua adoção como padrão na indústria automotiva, posição que mantém até hoje, segundo Buscemi et al. (2023), mesmo diante de alternativas mais modernas.

Com o avanço da conectividade veicular, conforme apontado por Lampe e Meng (2023), módulos externos de comunicação V2V e V2I passaram a se integrar aos sistemas internos dos veículos por meio de *gateways* que fazem a ponte entre a CAN e as redes externas, ampliando a superfície de ataque e introduzindo novos desafios de segurança.

2.2.1 Características e funcionamento

A rede CAN opera com base na transmissão de pacotes de dados organizados em quadros padronizados. Conforme detalhado por Buscemi et al. (2023), cada quadro inclui os seguintes campos: *Start of Frame* (SOF), que marca o início da transmissão; campo de arbitragem, que carrega o identificador de prioridade do quadro; campo de controle, que indica o comprimento dos dados; campo de dados, com 0 a 8 bytes de *payload*; *Cyclic Redundancy Check* (CRC), que detecta erros de transmissão; *Acknowledgement* (ACK), pelo qual os receptores confirmam o recebimento correto; e *End of Frame* (EOF), que indica o fim da mensagem. A Figura 3 ilustra essa estrutura.

O protocolo CAN define duas versões com tamanhos de identificador distintos. O

SoF	Arbitration Field	Control Field	Data Field	CRC	ACK	EoF	Inter Frame Space
1 bit	12 bits	6 bits	0 to 8 bytes	16 bits	2 bits	7 bits	3 bits

Figura 3 – Estrutura do quadro de dados CAN. Adaptado de (LOKMAN; OTHMAN; ABU-BAKAR, 2019).

CAN 2.0A utiliza identificadores de 11 bits (formato padrão), permitindo até 2.048 identificadores únicos. O CAN 2.0B estende esse campo para 29 bits (formato estendido), ampliando o espaço de endereçamento para mais de 536 milhões de identificadores. Identificadores menores têm maior prioridade no barramento, ou seja, quanto menor o valor numérico do ID, mais urgente é a mensagem.

Além disso, o protocolo abrange as duas primeiras camadas do modelo de referência *Open Systems Interconnection* (OSI). Na camada física (ISO 11898-2 e ISO 11898-3), o barramento utiliza um par de fios trançados denominados *CAN High* (CANH) e *CAN Low* (CANL). Há duas implementações principais: o CAN de alta velocidade (ISO 11898-2), padronizado por *International Organization for Standardization* (2003), com taxas de até 1 Mbit/s e topologia de barramento com terminações em ambas as extremidades; e o CAN de baixa velocidade tolerante a falhas (ISO 11898-3), padronizado por *International Organization for Standardization* (2006), com taxas de até 125 kbit/s e capacidade de continuar operando mesmo com falha em um dos fios.

Na camada de enlace de dados (ISO 11898-1), conforme descrito por Boland et al. (2021), a comunicação ocorre por meio de quadros que não incluem endereços de origem ou destino nas mensagens. Cada ECU conectada ao barramento recebe todos os quadros transmitidos e decide, com base no identificador, se deve ou não processar aquela mensagem. Quando múltiplas ECUs iniciam uma transmissão simultaneamente, o protocolo resolve o conflito por meio de arbitragem não destrutiva por conteúdo, em que cada nó transmite seu identificador bit a bit e monitora o barramento ao mesmo tempo. Como o bit dominante prevalece sobre o recessivo, o nó cujo identificador produz o primeiro bit recessivo enquanto o barramento permanece dominante detecta que perdeu a arbitragem e interrompe sua transmissão. O nó vencedor continua sem interrupção e sem necessidade de retransmissão, garantindo que mensagens com identificadores de menor valor sejam sempre entregues primeiro.

O barramento CAN opera em modo *broadcast*, ou seja, todo quadro transmitido é visível para todos os nós conectados. Essa característica, útil para a comunicação eficiente entre subsistemas, representa ao mesmo tempo uma vulnerabilidade de segurança: conforme demonstrado por Koscher et al. (2010), qualquer nó com acesso físico ao barramento

pode tanto monitorar todo o tráfego quanto injetar mensagens, sem que o protocolo disponha de mecanismos nativos de autenticação ou criptografia para verificar a legitimidade do transmissor.

A coleta de dados do barramento CAN geralmente é feita por meio de dispositivos chamados *loggers* CAN, conectados fisicamente ao barramento. Conforme descrito por Kang et al. (2018), esses dispositivos podem ser acoplados ao conector OBD-II, presente na maioria dos veículos modernos, permitindo a captura das mensagens transmitidas pelas ECUs. Essa coleta é fundamental para a criação de bases de dados para análise de segurança, embora o processo possa variar dependendo do modelo do veículo e de seus sistemas. Vale ressaltar que algumas montadoras têm implementado *gateways* e restrições entre o conector OBD-II e o barramento CAN principal, o que, segundo Awaad et al. (2023), representa um desafio adicional para a coleta de dados em contextos de pesquisa.

Os dados do barramento CAN são normalmente documentados em arquivos *Database CAN* (DBC), que, conforme apresentado por Buscemi et al. (2023), descrevem a posição, o significado, a unidade e o formato de cada sinal transmitido no barramento. Com um arquivo DBC, é possível converter os bytes brutos de *payload* em grandezas físicas legíveis, como velocidade em km/h ou rotação do motor em rpm, facilitando tanto a análise de segurança quanto o desenvolvimento de aplicações baseadas em CAN. No entanto, conforme observado por Marchetti e Stabili (2019), os arquivos DBC são proprietários de cada fabricante e raramente disponibilizados publicamente, o que torna a interpretação semântica do tráfego CAN um desafio adicional em pesquisas de segurança automotiva.

2.2.2 *Outros protocolos de comunicação veicular*

Apesar de sua ampla adoção, o protocolo CAN apresenta limitações crescentes frente às demandas dos veículos autônomos modernos, especialmente em termos de largura de banda e segurança. Para amenizar essas limitações, conforme aponta Buscemi et al. (2023), a indústria automotiva tem desenvolvido e adotado protocolos complementares e alternativos.

O *Local Interconnect Network* (LIN) é um protocolo de baixo custo e baixa velocidade utilizado em subsistemas de menor criticidade, como janelas elétricas, espelhos, assentos e iluminação. Por operar como rede secundária auxiliar, o LIN não substitui o barramento principal, mas complementa a arquitetura veicular ao assumir funções que não exigem alta taxa de transmissão, conforme descrito por Buscemi et al. (2023).

O *CAN with Flexible Data-Rate* (CAN-FD), introduzido pela Bosch em 2012, man-

tém compatibilidade com a infraestrutura física do CAN convencional, mas oferece, segundo Buscemi et al. (2023), taxas de transferência e capacidade de *payload* significativamente superiores. Essa evolução representa uma solução de transição adotada por diversas montadoras por não exigir a substituição completa do cabeamento existente.

O FlexRay foi projetado para aplicações de tempo real com requisitos elevados de determinismo e velocidade, conforme descrito por Lokman, Othman e Abu-Bakar (2019). Embora ofereça maior confiabilidade para sistemas críticos como controle de chassi e suspensão ativa, seu custo de implementação mais elevado limita sua popularidade.

A Ethernet automotiva, desenvolvida com base nas especificações IEEE 802.3, é, segundo Buscemi et al. (2023), a principal tecnologia adotada para comunicação de alta largura de banda entre unidades de processamento, câmeras e sensores LiDAR em veículos autônomos. No entanto, sua integração com os sistemas CAN já instalados ainda apresenta desafios de compatibilidade.

Apesar dessas alternativas, o CAN convencional e o CAN-FD permanecem como os padrões dominantes na maioria dos veículos em produção, apoiados pela grande quantidade de componentes compatíveis e pelo baixo custo de implementação. Por essas razões, segundo Buscemi et al. (2023), o protocolo CAN continua sendo o principal alvo de pesquisas em segurança automotiva.

2.2.3 Arquitetura de redes veiculares e vetores de ataque ao barramento CAN

Para entender como o barramento CAN pode ser alvo de ataques cibernéticos, é preciso primeiro compreender como ele está inserido na rede de comunicação de um veículo moderno e quais caminhos um invasor pode percorrer até alcançá-lo.

Um veículo moderno não é apenas um conjunto de peças mecânicas. Ele é controlado por dezenas de pequenos computadores embarcados chamados ECUs. Cada ECU é responsável por uma função específica: uma controla o motor, outra gerencia os freios, outra regula o ar-condicionado, e assim por diante. Conforme descrito por Buscemi et al. (2023), veículos modernos normalmente possuem entre 30 e 70 ECUs que trocam informações entre si de forma constante, e é exatamente para isso que a rede CAN existe — ela funciona como o principal meio de comunicação entre essas unidades.

No entanto, à medida que o número de ECUs cresceu, tornou-se necessário organizar a comunicação em grupos chamados domínios. Conforme descrito por Anwar et al. (2023), cada domínio reúne as ECUs de uma mesma área de responsabilidade e utiliza o protocolo de comunicação mais adequado às suas necessidades.

No núcleo do veículo estão os domínios mais críticos. O domínio de trem de força

controla motor e transmissão, enquanto o domínio de chassi é responsável por freios, direção e estabilidade. Ambos utilizam o CAN como protocolo principal, pois, segundo Buscemi et al. (2023), suas mensagens precisam ser entregues de forma rápida e confiável, mesmo que em volumes pequenos de dados.

Em camadas intermediárias estão o domínio de carroceria, que gerencia funções de conforto como janelas e iluminação por meio do protocolo LIN, e o domínio de *Advanced Driver Assistance Systems* (ADAS), que processa os dados de câmeras, *LiDAR* e radar utilizados nas funções de assistência à condução. Devido ao grande volume de dados gerado por esses sensores, o domínio de ADAS opera sobre Ethernet automotiva, capaz de transferências muito mais rápidas do que o CAN convencional (BUSCEMI et al., 2023). Nas camadas mais externas estão o domínio de infoentretenimento, que gerencia telas, navegação e integração com dispositivos do usuário, e o de telemática, responsável pela comunicação com servidores externos, outros veículos e a infraestrutura viária por redes celulares e V2X (ANWAR et al., 2023).

Os domínios não são redes completamente independentes, pois eles precisam trocar informações entre si. Essa conexão é feita por um componente chamado *gateway* central que, conforme descrito por Anwar et al. (2023), funciona como um roteador interno do veículo. O *gateway* recebe mensagens de um domínio, converte o formato quando necessário e as encaminha ao domínio de destino. É por meio do *gateway* que, por exemplo, uma informação de velocidade obtida pelo domínio de trem de força pode ser exibida no painel do infoentretenimento.

Do ponto de vista da segurança, o *gateway* representa ao mesmo tempo uma barreira e um ponto de risco. Como barreira, ele poderia, em tese, bloquear mensagens vindas de domínios periféricos antes que cheguem ao barramento CAN. Na prática, conforme apontado por Miller e Valasek (2014), as regras de filtragem do *gateway* são projetadas para garantir o funcionamento entre os domínios, e não para resistir a ataques. Isso significa que um invasor que consiga acessar qualquer ponto da rede interna pode usar o *gateway* para alcançar outros domínios, incluindo os que operam sobre o barramento CAN.

2.2.4 Interfaces externas e pontos de entrada

Além das redes internas, os veículos autônomos modernos possuem múltiplas interfaces que os conectam ao ambiente externo. Segundo Miller e Valasek (2014), essas interfaces ampliam as possibilidades de conectividade, mas também aumentam os pontos de entrada para possíveis ataques.

O conector OBD-II, localizado fisicamente no interior do habitáculo, foi criado

para diagnóstico e manutenção do veículo; porém, conforme demonstrado por Kang et al. (2018), qualquer dispositivo conectado a essa porta obtém acesso direto ao barramento CAN, podendo capturar e injetar mensagens sem restrições.

Além disso, as interfaces Bluetooth e Wi-Fi do domínio de infoentretenimento, voltadas para a comunicação com celulares e outros dispositivos, podem ter falhas exploradas por um invasor próximo ao veículo para obter acesso ao sistema interno (ANWAR et al., 2023). A conectividade celular, utilizada para atualizações remotas de *software* e diagnóstico à distância, representa, conforme demonstrado por Miller e Valasek (2015), o vetor de ataque remoto de maior alcance por não exigir proximidade física com o veículo.

Por fim, a comunicação V2X, que permite a troca de dados com outros veículos e com a infraestrutura viária, pode ser explorada por agentes maliciosos por meio de mensagens falsas que influenciem diretamente as decisões autônomas do veículo, como frenagens de emergência ou mudanças de trajetória, conforme apontado por Anwar et al. (2023).

2.2.5 Vulnerabilidades do barramento CAN

Apesar de sua eficiência operacional, o barramento CAN foi projetado em um contexto em que a segurança cibernética não era uma preocupação relevante, o que resultou em vulnerabilidades estruturais que persistem até hoje. Conforme apontado por Iehira, Inoue e Ishida (2018), a ausência de identificação de origem nos quadros significa que qualquer nó conectado ao barramento pode transmitir mensagens se passando por qualquer subsistema, sem que os demais nós tenham meios nativos de verificar a autenticidade do transmissor. Essa característica torna o protocolo suscetível a quatro categorias de ataque bastante conhecidas.

Os ataques de injeção de mensagens introduzem, conforme descrito por Cho e Shin (2016), quadros maliciosos no barramento para manipular o comportamento de ECUs específicas. Os ataques de *fuzzing*, segundo Buscemi et al. (2023), enviam mensagens com identificadores e *payloads* aleatórios ou malformados, com o objetivo de identificar falhas nos *parsers* das ECUs ou provocar comportamentos inesperados. Os ataques de negação de serviço sobrecarregam o barramento com transmissão massiva de mensagens de máxima prioridade, impedindo que as ECUs legítimas comuniquem-se entre si. Por fim, conforme descrito por Lee, Jeong e Kim (2017), os ataques de mascaramento fazem com que um nó malicioso assuma a identidade de um nó legítimo para transmitir comandos fraudulentos.

2.2.6 *Comunicação CAN em veículos autônomos*

Em veículos autônomos, a rede CAN assume um papel ainda mais crítico. Além de conectar os componentes tradicionais, integra também os sistemas ADAS e as unidades de processamento responsáveis pela tomada de decisões autônomas. Essa integração aumenta significativamente o volume e a frequência de tráfego no barramento.

Conforme observado por Lokman, Othman e Abu-Bakar (2019), os veículos autônomos apresentam padrões de tráfego no barramento CAN distintos dos veículos convencionais, com maior variação nos dados transmitidos e comportamentos específicos associados às funcionalidades autônomas. Durante manobras autônomas complexas, como mudança de faixa ou frenagem de emergência, ocorre um aumento expressivo no tráfego relacionado a sensores e atuadores, gerando padrões que podem ser interpretados incorretamente como anômalos por sistemas de detecção que não foram bem treinados.

Segundo Lee, Jeong e Kim (2017), o comportamento da rede CAN em veículos autônomos também varia de acordo com os diferentes modos de operação e condições ambientais. Além disso, o padrão de tráfego durante navegação autônoma em rodovia varia significativamente daquele observado em estacionamento autônomo ou em condições climáticas adversas.

2.2.7 *Como a rede CAN é comprometida na prática*

Após compreender a arquitetura, é possível entender como um ataque à CAN ocorre na prática. Como a rede não está exposta diretamente ao exterior, o invasor raramente a atinge de forma direta. O ataque se desenvolve em etapas, em que o invasor vai avançando pelas camadas da arquitetura, conforme ilustrado na Figura 1.

Em um cenário típico de ataque remoto, o invasor começa explorando uma falha em uma das interfaces externas do veículo, como o módulo de telemática celular ou o sistema de infoentretenimento. A partir desse ponto de entrada, o invasor passa a executar ações dentro do sistema comprometido e começa a se mover pela rede interna, passando de domínio em domínio por meio do *gateway*. Esse avanço continua até que o invasor alcance os domínios de trem de força ou chassi, onde o barramento CAN conecta as ECUs de controle físico. Uma vez no barramento, conforme demonstrado por Koscher et al. (2010), o invasor pode monitorar todo o tráfego e injetar mensagens que serão aceitas pelas ECUs, já que o barramento transmite as mensagens para todos os nós ao mesmo tempo.

Esse encadeamento foi demonstrado de forma concreta por Miller e Valasek (2015), que exploraram uma falha no módulo de telemática de um Jeep Cherokee para, sem qualquer acesso físico ao veículo, desativar motor, transmissão e freios enquanto o carro

estava em movimento. O caso mostra como a ausência de segurança na rede CAN se torna crítica quando combinada com a conectividade dos veículos autônomos, e fundamenta diretamente a necessidade de IDSs capazes de monitorar o tráfego de informações em diferentes contextos.

2.3 Sistemas de detecção de intrusão

Os *Intrusion Detection Systems* (IDSs) atuam na proteção das redes veiculares contra ataques cibernéticos. Esses sistemas são projetados para monitorar continuamente o tráfego de dados, identificar padrões que apresentem anomalias e responder a possíveis ameaças antes que causem danos significativos (ABDULGANIYU; TCHAKOUCHE; SAHEED, 2023).

2.3.1 Tipos de sistemas de detecção de intrusão

Existem três principais tipos de sistemas de detecção: baseados em assinaturas, baseados em anomalias e sistemas híbridos. As Figuras 4a e 4b exemplificam o funcionamento dos dois primeiros tipos. Os sistemas baseados em assinaturas funcionam por meio do armazenamento de um conjunto de padrões ou assinaturas de ataques conhecidos em um banco de dados (OTOUM; NAYAK, 2021). Quando novos pacotes de dados chegam à rede, suas características são comparadas com as assinaturas armazenadas. Essa abordagem é altamente eficiente na detecção de ataques previamente documentados, uma vez que a identificação ocorre de forma direta e precisa.

Além disso, a baixa taxa de falsos positivos torna esse método confiável em ambientes onde a maioria das ameaças já foi catalogada. No entanto, ele não é eficaz para detectar ataques desconhecidos, que não possuem assinaturas pré-registradas. Para manter sua eficácia, os sistemas baseados em assinaturas dependem de atualizações frequentes do banco de dados, o que pode ser desafiador em cenários de rápida evolução de ameaças.

Por outro lado, conforme descrito por Otoum e Nayak (2021), os IDSs baseados em anomalias monitoram continuamente o tráfego da rede e utilizam técnicas, como aprendizado de máquina e análise estatística, para identificar desvios em relação ao comportamento esperado (OTOUM; NAYAK, 2021). Esses sistemas analisam os atributos dos pacotes de dados, diferenciando o tráfego normal do tráfego malicioso com base em padrões aprendidos, sendo eficazes na detecção de ataques novos, que não possuem assinaturas conhecidas.

No entanto, as soluções baseadas em anomalias enfrentam desafios relacionados à precisão de seus algoritmos (MASEER et al., 2021). Um modelo ineficaz pode gerar falsos positivos, sinalizando atividades legítimas como ameaças, ou falsos negativos, deixando

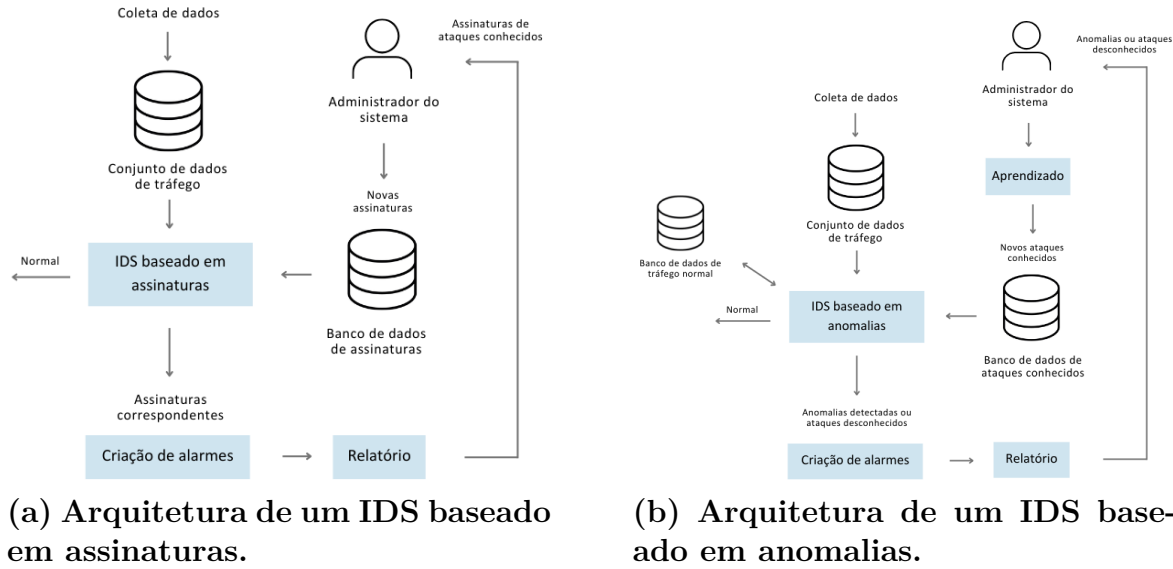


Figura 4 – Comparação entre as arquiteturas de *Intrusion Detection Systems* (IDSs) baseados em assinatura e baseados em anomalias. Adaptado de (BANGUI; GE; BUHNOVA, 2022).

passar ataques reais despercebidos. Isso pode sobrecarregar administradores de sistemas com alertas desnecessários ou comprometer a segurança da rede.

O desempenho desse tipo de sistema depende diretamente da qualidade dos algoritmos utilizados, da representatividade dos dados usados para treinamento e da capacidade do sistema de se adaptar a mudanças no comportamento da rede ao longo do tempo. Além disso, o processo de treinamento pode demandar grande esforço computacional.

Os sistemas híbridos, conforme apresentado por Bangui, Ge e Buhnova (2022), por sua vez, combinam as abordagens baseadas em assinaturas e em anomalias com o objetivo de aproveitar as vantagens de ambas e mitigar suas limitações individuais. Nessa arquitetura, o componente baseado em assinaturas atua como primeira camada de filtragem, identificando ameaças conhecidas com alta precisão e baixo custo computacional, enquanto o componente baseado em anomalias complementa a detecção ao identificar padrões suspeitos não catalogados. Essa abordagem tem atingido resultados promissores em cenários de redes veiculares, onde existe uma grande combinação de ataques conhecidos e variantes inédita.

2.3.2 Desafios na detecção de intrusão em redes CAN

A detecção de intrusão em redes CAN de veículos autônomos apresenta desafios únicos que derivam tanto das fragilidades inerentes ao protocolo quanto de aspectos práticos da implementação de segurança. Koscher et al. (2010) apresenta uma análise experimental

detalhada dessas vulnerabilidades, identificando diversas categorias de desafios.

Em primeiro lugar, a natureza de *broadcast* do protocolo CAN representa uma vulnerabilidade fundamental. Como os pacotes CAN são transmitidos tanto física quanto logicamente para todos os nós da rede, componentes maliciosos podem facilmente monitorar todas as comunicações ou enviar pacotes para qualquer outro nó. Essa característica facilita tanto a engenharia reversa do tráfego veicular quanto a injeção de mensagens maliciosas, dificultando significativamente a detecção de atividades ilegítimas.

Um segundo desafio crítico é a fragilidade do protocolo CAN a ataques de *Denial of Service* (DoS). Além de simples ataques de inundação de pacotes, o esquema de arbitragem baseado em prioridade do CAN permite que um nó afirme um estado dominante no barramento indefinidamente, fazendo com que todos os outros nós cedam a vez. Essa vulnerabilidade pode comprometer completamente a disponibilidade da rede, afetando funções críticas de segurança em veículos autônomos.

O terceiro desafio é a ausência de campos de autenticação ou identificação de origem nos pacotes CAN. Conforme detalhado por Koscher et al. (2010), isso significa que qualquer componente pode enviar indistinguívelmente um pacote a qualquer outro componente. Na prática, um único componente comprometido pode ser usado para controlar todos os outros componentes naquele barramento, desde que esses componentes não implementem defesas adicionais.

Os sistemas de controle de acesso implementados em veículos modernos também apresentam fragilidades relevantes. Lampe e Meng (2023) destacam que os mecanismos utilizados para proteger ECUs contra ações não autorizadas frequentemente empregam sementes e chaves de apenas 16 bits, tornando-os vulneráveis a ataques de força bruta.

Além das vulnerabilidades do protocolo, a variabilidade do comportamento normal da rede CAN em veículos autônomos constitui um desafio adicional. Conforme observado por Longari et al. (2020), a rede CAN apresenta padrões significativamente diferentes dependendo do estado de operação do veículo e do nível de autonomia ativado, o que pode levar a altas taxas de falsos positivos em sistemas de detecção que não consideram adequadamente essa variabilidade contextual.

Por fim, existe uma escassez de bases de dados abrangentes que representem adequadamente os diversos cenários de operação e de ataque em veículos autônomos. De acordo com Sanaboina, Naik e Rajiv (2023), conjuntos de dados limitados ou pouco representativos comprometem significativamente a eficácia dos algoritmos de detecção. A superação desses desafios exige o desenvolvimento de bases de dados que capturem a diversidade de comportamentos normais e anômalos nas redes CAN de veículos autônomos, bem como metodologias de detecção que possam se adaptar a essa variabilidade.

2.4 O papel das bases de dados no desenvolvimento de IDSs

O desenvolvimento de sistemas eficazes para detecção e avaliação de intrusões em redes CAN depende fundamentalmente da qualidade, abrangência e representatividade das bases de dados utilizadas para treinamento e validação (LAMPE; MENG, 2024b). Bases de dados inadequadas ou limitadas podem resultar em soluções de segurança que falham em identificar ataques em situações reais, gerando falsos positivos ou falsos negativos que comprometem a segurança dos veículos autônomos.

A representatividade do tráfego normal é particularmente desafiadora, pois, de acordo com Lokman, Othman e Abu-Bakar (2019), o comportamento normal de uma rede CAN pode variar significativamente dependendo das condições de operação do veículo, como aceleração, frenagem, manobras de estacionamento ou navegação autônoma em diferentes terrenos e condições climáticas.

Além disso, uma base de dados de qualidade deve incluir metadados contextuais que permitam relacionar o comportamento da rede CAN com o estado operacional do veículo (LEE; JEONG; KIM, 2017). Essa associação é especialmente importante em veículos autônomos, onde a interpretação dos padrões de comunicação depende do estado operacional do veículo.

2.4.1 *Heterogeneidade de cenários em segurança veicular*

A heterogeneidade de cenários em bases de dados para segurança veicular faz referência à diversidade de situações, condições operacionais e tipos de ataques representados. Lokman, Othman e Abu-Bakar (2019) argumenta que bases de dados com cenários diversos são essenciais para avaliar adequadamente a robustez dos IDSs, uma vez que atacantes reais podem adaptar suas técnicas a diferentes contextos.

Em veículos autônomos, a heterogeneidade dos dados assume importância ainda maior devido à complexidade dos sistemas e à variedade de situações que esses veículos podem enfrentar. Bases de dados que não contemplam essa diversidade podem resultar em sistemas de proteção que funcionam bem apenas em condições limitadas ou específicas, deixando vulnerabilidades em cenários não representados (CHO; SHIN, 2016).

2.4.2 *Desafios na geração de bases representativas*

A geração de bases de dados representativas para análise de intrusão em redes CAN enfrenta diversos desafios técnicos e práticos. O acesso limitado a veículos autônomos reais para coleta de dados, preocupações com segurança na simulação de ataques e a dificuldade de reproduzir a ampla gama de condições operacionais são obstáculos significativos

(KOSCHER et al., 2010).

Além disso, a categorização precisa dos dados, especialmente para a identificação de comportamentos anômalos sutis, a sincronização temporal entre diferentes fontes de dados e a padronização dos formatos também representam desafios técnicos relevantes (SEO; SONG; KIM, 2018). Conforme apresentado por Lampe e Meng (2024a), grande parte dos *datasets* públicos disponíveis apresenta limitações nessas dimensões, o que compromete sua aplicabilidade direta no desenvolvimento de IDSs.

Para superar esses obstáculos, pesquisadores têm adotado abordagens híbridas que combinam dados reais com simulações, permitindo a criação de bases mais abrangentes sem comprometer a segurança (TAYLOR; LEBLANC; JAPKOWICZ, 2016), mas ainda há desafios. A disponibilização pública de bases de dados representativas é pouco comum, porém representa uma contribuição significativa para o avanço da pesquisa em segurança de veículos autônomos (LAMPE; MENG, 2024b).

3 TRABALHOS RELACIONADOS

Este capítulo analisa os principais trabalhos relacionados ao desenvolvimento e uso de bases de dados para análise de intrusão em redes CAN de veículos autônomos, identificando contribuições relevantes e limitações que esta pesquisa busca superar.

3.1 *Datasets* para detecção de intrusão veicular

A disponibilidade de bases de dados representativas é fundamental para desenvolver sistemas eficazes de detecção de intrusão em redes CAN. Diversos pesquisadores têm contribuído nessa área, embora ainda existam falhas importantes, especialmente no contexto de veículos autônomos.

Antes de comparar os trabalhos existentes, é necessário definir os critérios utilizados para classificar a representatividade de cada base de dados. Para isso, foi realizada a construção da escala a seguir abaixo. A mesma foi utilizada para classificar os trabalhos relacionados encontrados.

- **Baixa:** base coletada em um único modelo de veículo convencional, cobrindo um único cenário operacional (por exemplo, veículo parado ou condução normal sem variação de via);
- **Média:** base que atende a pelo menos um dos seguintes critérios: (i) dois ou mais modelos de veículos distintos; (ii) múltiplos cenários de via (por exemplo, urbano e rodovia); (iii) quatro ou mais tipos de ataque documentados;
- **Alta:** base que satisfaz todos os critérios de nível médio e incorpora, adicionalmente, variação sistemática de condições ambientais e climáticas em cenários de operação autônoma, com metadados contextuais que permitem rastrear cada registro até seu contexto de origem.

O desenvolvimento de conjuntos de dados para análise de segurança veicular tem evoluído nos últimos anos. Song, Kim e Kim (2016) são pioneiros por apresentarem uma das primeiras bases para detecção de anomalias em redes CAN, baseada na análise de intervalos de tempo entre mensagens, evidenciando a eficácia de características temporais para identificar comportamentos anômalos, porém explorando apenas cenários de direção convencional. De forma similar, Lee, Jeong e Kim (2017) expandem essa abordagem ao

proporem o *On-board Time-based Intrusion Detection System* (OTIDS), um sistema que usa quadros remotos CAN para detecção de intrusão, junto com uma base de dados que inclui cenários de ataques de DoS e *fuzzing* coletados em veículo real.

A necessidade de bases de dados mais completas é destacada por Avatefipour e Malik (2018) em sua revisão sobre vulnerabilidades em redes CAN. Os autores apontam uma limitação comum à maioria dos *datasets* disponíveis que é a cobertura restrita a poucos cenários operacionais e tipos de ataque, que não reflete a variedade de situações encontradas na prática. Essa observação é especialmente relevante para veículos autônomos, cujos padrões de comunicação interna são mais complexos e variáveis que os de veículos convencionais.

O conjunto de dados ROAD (*Real ORNL Automotive Dynamometer*), proposto por Dupont et al. (2019), representa um avanço significativo ao ser coletado em três modelos de veículos reais distintos (Opel Astra, Renault Clio e Toyota Corolla) com múltiplos cenários controlados de ataque, incluindo *injection*, *masquerade* e DoS.

Apesar da maior diversidade de veículos em relação aos trabalhos anteriores, o *dataset* ainda se restringe a condições de condução convencional, sem contemplar funcionalidades autônomas. Em veículos autônomos, o barramento CAN passa a transportar também comandos gerados automaticamente, por exemplo controle de trajetória e frenagem autônoma o que amplia a superfície de ataque e torna insuficientes os *datasets* coletados exclusivamente em condições de condução convencional para o treinamento de IDSs.

O Car-Hacking Dataset (SEO; SONG; KIM, 2018), coletado em veículo real via conector OBD-II, totaliza mais de 17 milhões de registros CAN abrangendo quatro tipos de ataque, sendo eles: DoS, *Fuzzing*, *RPM Spoofing* e *Gear Spoofing*, além de tráfego normal. Apesar do volume expressivo de registros e da cobertura de múltiplos tipos de ataque, a base se restringe a um único veículo e a condições de condução convencional, sem variação de cenários operacionais.

Um avanço mais recente é representado pelo *can-train-and-test* (LAMPE; MENG, 2024b), um *dataset* bem documentado que padroniza a estrutura de coleta e rotulagem para facilitar o uso em aprendizado de máquina supervisionado. Os autores destacam que grande parte dos *datasets* públicos anteriores necessitam de rotulagem adequada em nível de *frame*, o que compromete diretamente a reprodutibilidade e a comparação entre sistemas de detecção. Com cerca de 40 milhões de registros, é uma ordem de grandeza maior que as bases do laboratório *Hacking and Countermeasure Research Lab* (HCRL) (LEE; JEONG; KIM, 2017; SEO; SONG; KIM, 2018), oferecendo volume adequado para modelos de aprendizado profundo. Ainda assim, o *dataset* não contempla cenários de operação autônoma.

A Tabela 1 apresenta uma comparação entre as principais bases de dados utilizadas para detecção de intrusão em redes CAN veiculares e a proposta da presente pesquisa.

Tabela 1 – Comparação entre bases de dados para detecção de intrusão em redes CAN. A proposta destaca-se pela operação autônoma, variação sistemática de via e clima, e por incluir metadados contextuais (15 atributos).

Identificação	Represent.	Ataques	Ambiente	Veículos	Registros	Atrib.	Tam.	Cenários
(LEE; JEONG; KIM, 2017)	Baixa	DoS, <i>Fuzzing</i> , Impersonação	Real	Único modelo + Convencio- nal	~7 M	11	~490 MB	Único (condução normal)
(KANG et al., 2018)	Média	Injection, DoS, <i>Replay</i>	Real	Único modelo + Convencio- nal	~17 M	~11	~1,2 GB	Múltiplos (urbano, rodovia)
(DUPONT et al., 2019)	Média	Injection, Masquerade, DoS	Real	3 modelos + Convencio- nais	~12 M	~11	~850 MB	Múltiplos (controlados)
Car-Hacking (SEO; SONG; KIM, 2018)	Baixa	DoS, <i>Fuzzing</i> , <i>RPM Spoofing</i> , <i>Gear Spoofing</i>	Real	Único modelo + Convencio- nal	~17 M	11	~1,7 GB	Limitados (veículo parado)
can-train-and-test (LAMPE; MENG, 2024b)	Média	DoS, <i>Fuzzing</i> , Injection, <i>Spoofing</i> (9 tipos)	Real	4 modelos, 2 fabricantes + Convencio- nais	~130 M	12	~9 GB	Múltiplos (condução normal e em movimento)
Proposta da pesquisa	Alta	DoS, <i>Fuzzing</i>, <i>RPM</i> <i>Spoofing</i>, <i>Gear</i> <i>Spoofing</i>	Simulado	Múltiplos + Autôno- mos	~5,4 M / 3,6 M bal.	15	~490 MB / 280 MB bal.	45 cenários: urbano, rural, rodovia × seco, chuva, neve

Os *datasets* construídos pelo laboratório *Hacking and Countermeasure Research Lab* (HCRL) utilizam 11 atributos CAN 2.0B padrão (*Timestamp*, *CAN_ID*, *DLC*, *DATA[0-7]*, *Flag*) (SEO; SONG; KIM, 2018; LEE; JEONG; KIM, 2017); o can-train-and-test acrescenta *Label* e *SubClass* (12 atributos) (LAMPE; MENG, 2024b); a proposta adiciona *attack_type*, *road_type* e *climate* (15 atributos).

Como se pode observar na Tabela 1, as bases de dados existentes apresentam limitações significativas em termos de heterogeneidade, representação de veículos autônomos e diversidade de cenários operacionais. O *dataset* proposto neste trabalho é o único que combina variação sistemática de ambiente e condição climática em uma base voltada para segurança de redes CAN.

3.1.1 *Waymo Open Dataset*

O *Waymo Open Dataset* (Waymo LLC, 2019) é frequentemente citado em pesquisas sobre veículos autônomos. Trata-se de uma base de dados de percepção ambiental, composta por nuvens de pontos de cinco sensores LiDAR, imagens de cinco câmeras de alta resolução e trajetórias de objetos, coletados por frotas de veículos autônomos da Waymo em cidades norte-americanas (Waymo LLC, 2019). Sua finalidade é o desenvolvimento e a avaliação de algoritmos de detecção de objetos, rastreamento e predição de movimento.

O *Waymo* não contém dados de barramento CAN, não registra o tráfego de comunicação entre ECUs e não inclui cenários de ataque cibernético. Seu domínio é a percepção do ambiente externo ao veículo, enquanto o presente trabalho foca na segurança da rede de comunicação interna. Por essa razão, o *Waymo* não é comparável às bases listadas na Tabela 1 e não seria adequado para o treinamento ou avaliação dos IDSs aqui desenvolvidos, independentemente de sua qualidade e abrangência para veículos autônomos.

3.2 Abordagens de detecção e classificação de intrusões

Os métodos de análise de dados CAN e avaliação de sistemas de detecção têm se diversificado, incluindo desde técnicas estatísticas até abordagens de aprendizado de máquina. Essas pesquisas mostram tanto avanços importantes quanto desafios persistentes na segurança de redes veiculares.

As abordagens baseadas em características estatísticas do tráfego CAN foram as primeiras na área de detecção de intrusões. Müter e Asaj (2011) apresentam a eficácia de métricas de entropia para identificar anomalias na comunicação veicular, criando uma base importante para métodos de detecção não supervisionados. Embora promissora, essa técnica é testada apenas em ambientes controlados e com poucos cenários de ataque, e os autores não reportam métricas formais de desempenho como *F1-score*, deixando dúvidas sobre seu comportamento em condições mais variadas, como as encontradas em veículos autônomos.

O cenário evolui rapidamente com a aplicação de técnicas de aprendizado de máquina. Kang e Kang (2016) desenvolvem um sistema baseado em redes neurais profundas para detecção de intrusões em redes veiculares, atingindo acurácia superior a 99% na detecção de ataques de injeção de mensagens no *dataset* disponibilizado pelo laboratório *Hacking and Countermeasure Research Lab* (HCRL). Os próprios autores, contudo, reconhecem que o desempenho do sistema depende diretamente da qualidade e representatividade dos dados de treinamento, destacando a conexão direta entre a qualidade das bases disponíveis e a eficácia dos métodos de detecção.

Com o avanço das arquiteturas de aprendizado profundo, Longari et al. (2020) propõem o CANnolo, um sistema baseado em *LSTM autoencoders* treinado exclusivamente com dados normais do barramento CAN, sem necessidade de dados rotulados de ataque. Avaliado em três tipos de ataque, o sistema alcança taxas de detecção entre 91% e 100%, evidenciando que abordagens não supervisionadas podem ser eficazes para redes CAN. No entanto, os autores apontam que o alto custo computacional do modelo representa um desafio para implantação em ambientes embarcados com recursos limitados.

Em abordagem supervisionada, Hossain et al. (2020) propõem um IDS baseado em *Long Short-Term Memory* (LSTM) para detecção de ataques DoS, *fuzzing* e *spoofing* no barramento CAN, reportando alta taxa de detecção o *dataset* construído pelo *Hacking and Countermeasure Research Lab* (HCRL) (LEE; JEONG; KIM, 2017; SEO; SONG; KIM, 2018). Um diferencial do trabalho é que o modelo opera sobre os dados brutos do CAN sem necessidade de decodificação prévia, o que o torna aplicável a qualquer modelo de veículo. Os autores identificam, contudo, limitações na detecção de ataques com baixa taxa de injeção, situação em que a diferença entre tráfego normal e malicioso é mais sutil.

Além disso, um panorama das abordagens baseadas em inteligência artificial para detecção de intrusão em redes CAN é apresentado por Rajapaksha et al. (2023), que revisam sistematicamente trabalhos publicados entre 2016 e 2022. O *survey* mapeia técnicas como *Deep Neural Network* (DNN), LSTM, CNN, *autoencoders* e *Random Forest*, identificando as métricas utilizadas em cada trabalho, predominantemente acurácia, precisão, revocação e *F1-score*, observando que poucos estudos avaliam seus modelos em mais de um *dataset*, o que limita a generalização dos resultados. Os autores concluem que a dependência de *datasets* restritos e pouco heterogêneos é o principal fator limitante para o avanço dos IDSs em redes veiculares.

A Tabela 2 resume as principais características metodológicas dos trabalhos discutidos nesta seção e os contrasta com a abordagem adotada neste trabalho.

3.3 Lacunas identificadas e posicionamento desta pesquisa

A análise dos trabalhos relacionados mostra avanços significativos na área de segurança para redes CAN, mas também revela limitações que esta pesquisa busca preencher de forma direta.

Do ponto de vista das bases de dados, nenhum dos *datasets* analisados contempla dados de operações autônomas, como navegação em rodovia sem intervenção humana, manobras em condições climáticas adversas ou comportamentos específicos de controladores autônomos. Essa ausência compromete a validade dos IDSs treinados com esses dados quando aplicados a veículos autônomos, cujos padrões de comunicação CAN são

Tabela 2 – Comparação entre abordagens metodológicas para análise e avaliação de IDSs em redes CAN, destacando algoritmos utilizados, métricas reportadas e protocolo de validação. A proposta deste trabalho diferencia-se pela avaliação comparativa de quatro algoritmos com otimização automática via Optuna e validação cruzada em 45 cenários.

Trabalho	Algoritmo(s)	Métricas reportadas	Protocolo de validação
(MÜTER; ASAJ, 2011)	Entropia estatística	Qualitativa	Ambiente controlado único
(KANG; KANG, 2016)	DNN	Acurácia	Treino/teste simples
(LONGARI et al., 2020)	LSTM <i>Autoencoder</i>	Taxa de detecção	Treino/teste simples
(HOSSAIN et al., 2020)	LSTM	Acurácia, Precisão, <i>Recall</i>	Treino/teste simples
Proposta	Árvore de Decisão, <i>Random Forest</i>, XGBoost, CNN	F1-score, Precisão, <i>Recall</i>	<i>k-fold</i> por cenário ($k = 5$); otimização via Optuna/TPE

diferentes dos veículos convencionais (TAYLOR; LEBLANC; JAPKOWICZ, 2016).

Embora os tipos de ataque simulados neste trabalho sejam os mesmos presentes nos *datasets* do laboratório *Hacking and Countermeasure Research Lab* (HCRL) (LEE; JEONG; KIM, 2017; SEO; SONG; KIM, 2018), o diferencial da base proposta está na diversidade de contextos operacionais em que esses ataques são injetados. Essa combinação não é encontrada entre os trabalhos revisados e permite avaliar como o impacto de um mesmo ataque varia conforme o contexto operacional.

Do ponto de vista metodológico, Wang e Sawhney (2020) observam que a maioria dos conjuntos disponíveis foi coletada em ambientes controlados, sem refletir a variabilidade de condições encontradas em situações reais de uso. A presente pesquisa adota uma abordagem baseada em simulação de alta fidelidade no CARLA, combinando múltiplos cenários operacionais e permitindo reprodutibilidade completa sem os custos e riscos da coleta em veículos reais.

Por fim, conforme apontado por Rajapaksha et al. (2023), a avaliação entre diferentes *datasets* é prática rara na literatura, limitando a comparabilidade entre métodos. Este trabalho utiliza as bases desenvolvidas para realizar uma avaliação comparativa de múltiplos algoritmos de classificação.

Como evidenciado na Tabela 1, este trabalho se diferencia por oferecer alta representatividade, incluir cenários específicos de operação autônoma e contemplar uma ampla variedade de condições ambientais e tipos de ataque, respondendo às principais limitações

identificadas na literatura revisada.

Do ponto de vista metodológico, a Tabela 2 complementa essa análise ao mostrar que os trabalhos revisados, em sua maioria, avaliam um único algoritmo com métricas restritas e protocolos de validação simples. Este trabalho avança ao utilizar quatro algoritmos distintos para avaliar o conjunto de dados proposto.

4 MATERIAIS E MÉTODOS

Este capítulo descreve a abordagem metodológica para a criação e análise de bases de dados com cenários heterogêneos para redes CAN em veículos autônomos, desde a simulação de cenários diversos até a avaliação de algoritmos para detecção de intrusões. A metodologia integra simulações no CARLA, transformação dos dados em mensagens CAN, inserção controlada de ataques cibernéticos e preparação das bases para experimentos de classificação.

A Figura 5 apresenta uma visão geral do *pipeline* metodológico adotado neste trabalho, ilustrando o fluxo completo desde a simulação dos cenários até a avaliação dos algoritmos de detecção.

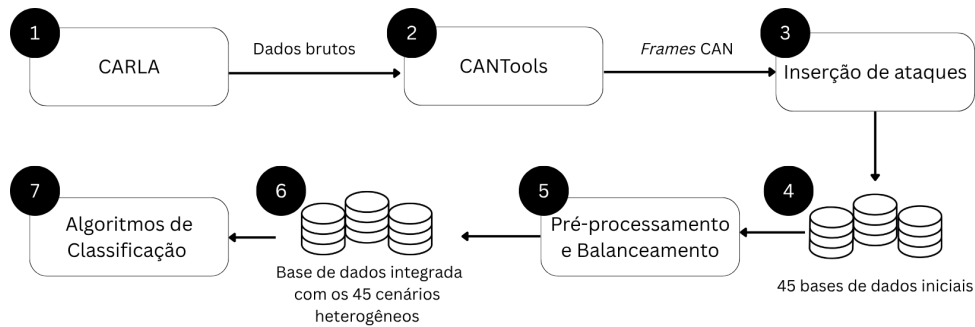


Figura 5 – Etapas metodológicas para elaboração do conjunto de dados. O fluxo parte da simulação no CARLA e percorre a conversão para o formato CAN, a injeção controlada de ataques, o pré-processamento e balanceamento, até a avaliação comparativa dos algoritmos de classificação.

As etapas principais são:

1. **Simulação no CARLA:** geração de trajetórias veiculares em três ambientes e três condições climáticas, produzindo dados brutos de operação normal do veículo autônomo.
2. **Transformação em *frames* CAN:** conversão dos dados simulados para o formato de mensagens CAN por meio da biblioteca *CANTools*, com mapeamento de sinais via arquivo *.dbc*.
3. **Injeção de ataques:** inserção controlada de eventos maliciosos (*DoS*, *Fuzzing*, *RPM Spoofing* e *Gear Spoofing*), gerando versões rotuladas para aprendizado supervisionado.

4. **Organização em 45 conjuntos de dados:** reorganização dos dados em todas as combinações possíveis de via, clima e categoria de mensagem (Normal, DoS, *Fuzzing*, *RPM Spoofing* e *Gear Spoofing*).
5. **Pré-processamento:** tratamento do desbalanceamento de classes via combinação de *undersampling* das classes majoritárias e *Synthetic Minority Over-sampling Technique* (SMOTE) nas classes minoritárias, remoção de registros inconsistentes e codificação de variáveis categóricas.
6. **Organização de uma base de dados única com cenários heterogêneos:** unificação das 45 bases em apenas um *dataset*.
7. **Treinamento dos algoritmos de classificação:** aplicação de algoritmos de *Decision Tree*, *Random Forest*, *XGBoost* e CNN, com otimização automática de hiperparâmetros (detalhada no Capítulo 6), e avaliação dos resultados obtidos por meio de métricas pré-estabelecidas.

As seções seguintes detalham cada uma dessas etapas.

4.1 Utilização de Inteligência Artificial

Os algoritmos de *machine learning* e scripts utilizados neste trabalho foram desenvolvidos manualmente. Após a redação manual dos códigos, foi utilizada a ferramenta de Inteligência Artificial Generativa (IAG) Copilot*, com o objetivo de corrigir erros sintáticos e organizar e documentar o código para garantir legibilidade e fácil compreensão.

Vale destacar que todas as escolhas metodológicas e os resultados obtidos são de responsabilidade da autora e que o uso da IAG foi restrito a essas etapas listadas anteriormente.

4.2 Construção e organização das bases de dados

A coleta dos dados foi realizada por meio de simulações no CARLA, onde foram executadas múltiplas trajetórias em três ambientes (urbano, rural e rodovia) sob três condições climáticas distintas (seco, chuva com neblina e neve). Em uma primeira fase foram gerados nove conjuntos brutos, variando via e clima. Após isso, cada conjunto recebeu injeções controladas de eventos maliciosos (DoS, *RPM Spoofing*, *Gear Spoofing*, *Fuzzing*) além de registros normais.

*Disponível em: <<https://copilot.microsoft.com/>>

Para facilitar análises comparativas e isolar o efeito de cada ataque, a base foi reorganizada em 45 conjuntos de dados, correspondendo a todas as combinações possíveis entre ambiente (3 cenários), condição climática (3 cenários) e categoria de mensagem (5 cenário). As cinco categorias de mensagem são: Normal (tráfego legítimo sem anomalias), DoS (sobrecarga do barramento), *Fuzzing* (mensagens com dados aleatórios ou malformados), *RPM Spoofing* (falsificação de mensagens de rotação do motor) e *Gear Spoofing* (falsificação de mensagens do sistema de transmissão).

A consolidação dos dados foi realizada por meio de um *script* de mesclagem, que produz o arquivo combinado. Esse arquivo agrega todas as amostras mantendo as colunas de metadados (cenário, clima, ambiente e rótulo de ataque), o que permite tanto análises globais quanto estudos por subgrupo.

4.2.1 Simulação de cenários com CARLA

O *Car Learning to Act* (CARLA) é um simulador de código aberto desenvolvido para pesquisa em sistemas de condução autônoma (DELAVARI; KHANZADA; KWON, 2025). A escolha do CARLA diante de outros simuladores, como SUMO, AirSim, Gazebo e SVL, ocorreu devido à sua capacidade de reproduzir diferentes níveis de autonomia veicular e variados contextos operacionais com alta fidelidade, sendo reconhecido como o mais eficaz para cenários de tráfego urbano em comparação com as demais alternativas disponíveis (LI et al., 2024). Além disso, o CARLA oferece controle sobre parâmetros ambientais como precipitação, neblina e aderência da pista, o que permite reproduzir cenários bem próximos da realidade (DELAVARI; KHANZADA; KWON, 2025).

Para garantir a diversidade destacada como limitação nos trabalhos de Wang e Sawhney (2020) e Taylor, Leblanc e Japkowicz (2016), foram simulados múltiplos cenários que incluem diferentes ambientes como urbano, rodovia e estrada rural, além de variadas condições climáticas como seco, chuva com neblina e neve. Os experimentos contemplaram também múltiplos níveis de autonomia, desde condução assistida até autonomia total, bem como diversas manobras autônomas como mudança de faixa, estacionamento e frenagem de emergência.

Por meio do simulador, obtêm-se dados referentes aos principais sistemas veiculares, incluindo estado do motor, sistema de transmissão, sistema de freios, direção e sensores ambientais. Estes dados representam o comportamento normal do veículo autônomo em diferentes contextos operacionais.

4.2.2 *Caracterização dos cenários climáticos e parâmetros de velocidade*

Com o objetivo de ampliar a representatividade dos cenários simulados e permitir a análise de comportamentos anômalos também sob condições ambientais adversas, foi incorporado um conjunto de parâmetros que consideram o ambiente, as condições climáticas e os limites de velocidade esperados para cada situação. Essa caracterização complementa a simulação realizada no CARLA, permitindo avaliar como diferentes fatores externos influenciam a operação veicular e o comportamento das mensagens CAN.

Foram definidos três ambientes: urbanos, estradas rurais de pista simples e rodovias. Cada um desses cenários apresenta dinâmicas distintas de tráfego, restrições normativas diferentes e exigências específicas quanto à velocidade de operação. Em seguida, para cada via, foram aplicadas três condições climáticas: clima seco, chuva com neblina e neve. Essas condições foram escolhidas por representarem situações frequentes em ambientes urbanos e rodoviários, além de afetarem diretamente sensores, frenagem, visibilidade e aderência ao solo.

A velocidade foi tratada como um parâmetro crítico de segurança, uma vez que pequenas variações podem alterar significativamente a dirigibilidade e o risco operacional. Assim, para cada combinação entre via e clima, foram estabelecidos limites máximos e mínimos de velocidade. Em condições adversas, como chuva intensa ou neve, os limites de velocidade foram reduzidos para refletir o comportamento esperado de um sistema autônomo capaz de se adaptar às condições de segurança do ambiente.

A definição dos limites de velocidade considera o Código de Trânsito Brasileiro (CTB), que estabelece reduções obrigatórias em situações de baixa visibilidade e pistas molhadas (BRASIL, 1997). Associado a isso, estudos sobre velocidade e segurança mostram que excessos mesmo moderados aumentam drasticamente o risco de incidentes (SINGH, 2015). Da mesma forma, pesquisas em controle autônomo comprovam a necessidade de ajustes dinâmicos de velocidade sob condições ambientais variáveis (ZHU et al., 2020; TAHIR et al., 2024).

Além dos limites regulamentares, também foram definidos um conjunto de regras para caracterização de comportamento anômalo. A Tabela 3 apresenta o conjunto de parâmetros adotados nos cenários analisados. Vale observar que, para a estrada rural, os limites de velocidade foram mantidos constantes entre as condições climáticas distintas do seco, uma vez que o perfil de tráfego de baixa densidade e a dinâmica de condução nesses ambientes são pouco afetados por reduções moderadas de visibilidade, diferentemente das rodovias, onde altas velocidades amplificam significativamente o risco em condições adversas.

Tabela 3 – Limites de velocidade mínima e máxima adotados para cada combinação de ambiente e condição climática nos cenários simulados, definidos com base no Código de Trânsito Brasileiro (CTB) e na literatura sobre controle autônomo em condições adversas. Reduções de velocidade em chuva e neve refletem o comportamento esperado de um sistema autônomo adaptativo.

Cenário	Velocidade Máx. (km/h)	Velocidade Mín. (km/h)	Clima
Urbano	60	30	Seco
Rural	60	30	Seco
Rodovia	120	60	Seco
Urbano	50	30	Chuva + Neblina
Rural	60	30	Chuva + Neblina
Rodovia	90	60	Chuva + Neblina
Urbano	50	30	Neve
Rural	60	30	Neve
Rodovia	90	60	Neve

4.2.3 Transformação em frames CAN com CANTools

Os dados gerados pelo simulador foram convertidos para o formato de *frames* CAN utilizando a biblioteca *CANTools* (MOQVIST, 2024). Essa ferramenta foi escolhida por sua documentação completa e suporte nativo ao formato *.dbc*, que é o padrão industrial para descrição de sinais CAN, permitindo que a estrutura de dados gerada seja compatível com ferramentas de análise utilizadas na indústria automotiva.

O processo de conversão seguiu etapas estruturadas, iniciando com a definição de uma base de dados CAN por meio da criação de um arquivo *.dbc* específico para mapear os sinais do simulador em mensagens CAN. Em seguida, definiram-se identificadores CAN para cada subsistema veicular, formato e tamanho de cada sinal, seguido pelo mapeamento dos dados brutos do simulador para os respectivos sinais CAN. Por fim, realizou-se a conversão em *frames* CAN completos, incluindo CAN ID, *Data Length Code* (DLC), campos de dados e *timestamp*.

Este processo permitiu a criação de dados CAN que refletem as características temporais e funcionais identificadas como relevantes por Song, Kim e Kim (2016) e Mütter e Asaj (2011) em seus estudos sobre detecção de anomalias.

4.2.4 Injeção e modelagem dos cenários de ataque

Para criar bases de dados que representassem tanto comportamentos normais quanto anômalos, foram introduzidos *frames* CAN maliciosos seguindo variados padrões de ataque. Vale destacar que seleção dos ataques descritos nesta seção não foi aleatória. Cada um deles corresponde a categorias de ameaças já discutidas em trabalhos relacionados e sintetizadas na Tabela 1. Sendo assim, implementaram-se quatro categorias de ataque descritas a seguir.

O ataque de DoS consiste na injeção massiva de mensagens com identificadores de alta prioridade (IDs dominantes) em diferentes frequências, com o objetivo de saturar o barramento CAN e impedir a transmissão de mensagens legítimas. Este ataque explora diretamente o mecanismo de arbitragem do protocolo, no qual mensagens com IDs mais baixos prevalecem sobre as demais, podendo interromper funções críticas de segurança do veículo.

O ataque de *Fuzzing*, por sua vez, caracteriza-se pela introdução de mensagens com IDs e valores de *payload* aleatórios ou malformados. O objetivo é induzir comportamentos imprevisíveis nas ECUs ou revelar falhas de tratamento de exceções nos sistemas veiculares. Por não seguir padrões definidos, esse tipo de ataque é particularmente difícil de detectar por sistemas baseados em assinaturas.

Outro ataque conhecido, o *RPM Spoofing* trata-se de um ataque de falsificação direcionado ao subsistema de rotação do motor. Mensagens falsas são injetadas com valores de *Revolutions Per Minute* (RPM) adulterados, simulando condições de operação inexistentes. Podendo interferir em sistemas de controle de tração, câmbio automático e limitadores de rotação, comprometendo o comportamento dinâmico do veículo.

Por fim, o *Gear Spoofing* de forma semelhante ao *RPM Spoofing*, este ataque falsifica as mensagens associadas ao sistema de transmissão, injetando informações incorretas sobre a marcha. Essa manipulação pode comprometer o sistema de controle de estabilidade e as estratégias de frenagem em veículos elétricos e híbridos.

Todos os eventos maliciosos foram inseridos com rotulagem correta dos *frames* afetados, garantindo a rastreabilidade necessária para o treinamento supervisionado dos modelos de detecção. Cada *frame* recebe os atributos *Flag* (normal ou malicioso) e *Attack_Type* (categoria do ataque), conforme detalhado na Seção 4.3.

4.3 Pré-processamento dos dados

As bases de dados foram estruturadas com atributos que capturam não apenas as características básicas das mensagens CAN, mas também informações contextuais que

permitem análises mais sofisticadas. Os atributos de cada registro incluem: *Timestamp* (instante de captura do *frame*), CAN ID (identificador do subsistema transmissor), DLC (comprimento do campo de dados), DATA[0] a DATA[7] (bytes de *payload*), além dos indicadores de anomalia *Flag* (normal ou malicioso) e *Attack_Type* (categoria do ataque).

A distribuição inicial dos dados apresentou uma maior quantidade de registros maliciosos em relação aos registros normais. Esse comportamento é esperado em ambientes de teste de estresse, onde os cenários de ataque são incluídos em excesso para garantir volume suficiente de exemplos por categoria. A Tabela 4 exemplifica essa distribuição antes da aplicação de qualquer técnica de balanceamento.

Tabela 4 – Distribuição de registros por categoria antes do balanceamento no conjunto de treinamento bruto. Observa-se a predominância de registros de ataque (aproximadamente 81%) em relação aos normais (aproximadamente 19%). Esse desequilíbrio motivou a aplicação de *undersampling* combinado com Synthetic Minority Over-sampling Technique (SMOTE).

Categoria	Proporção aproximada (%)	Técnica de balanceamento
DoS	~39	<i>undersampling</i>
<i>Fuzzing</i>	~22	<i>undersampling</i>
<i>RPM Spoofing</i>	~10	SMOTE
<i>Gear Spoofing</i>	~9	SMOTE
Normal	~19	SMOTE

Para corrigir esse desbalanceamento exclusivamente no conjunto de treinamento, aplicou-se uma combinação de *undersampling* das classes majoritárias com a técnica SMOTE (CHAWLA et al., 2002) nas classes minoritárias. O *undersampling* reduz as classes com maior volume de registros, enquanto o SMOTE gera exemplos sintéticos das classes com menor representação por interpolação no espaço de características dos vizinhos mais próximos, aumentando sua diversidade sem introduzir *overfitting* por repetição de instâncias. O resultado dessa combinação foi uma distribuição uniforme, com cerca de 511.225 instâncias por categoria, totalizando aproximadamente 2,56 milhões de registros no conjunto de treinamento balanceado. O conjunto de teste foi mantido com a distribuição original, garantindo que a avaliação dos modelos reflita as condições reais de operação sem introduzir viés nas métricas reportadas.

A divisão dos dados em conjuntos de treinamento (80%) e teste (20%) foi realizada com base em cenários completos, em vez de uma separação aleatória por registro. Isso significa que todos os registros pertencentes a um mesmo cenário foram mantidos juntos em uma única partição. Essa decisão é importante porque, em dados de redes CAN, registros próximos no tempo tendem a apresentar padrões semelhantes. Caso a divisão fosse feita de forma aleatória, é possível que registros muito parecidos apareçam simultaneamente

nos conjuntos de treino e teste, o que pode favorecer o modelo durante a avaliação. Esse fenômeno é conhecido como *data leakage*, ou vazamento de dados, onde ao ser exposto a amostras estatisticamente similares às do teste, o modelo aprende padrões específicos daquelas instâncias em vez de generalizar o comportamento do barramento, resultando em métricas de desempenho otimistas que não se confirmam em cenários reais.

4.3.1 Limpeza e codificação dos dados

Instâncias com campos ausentes ou com inconsistências, como valores de DLC fora do intervalo válido (0 a 8 bytes) ou *timestamps* fora de ordem, foram identificados e removidos antes da etapa de modelagem. Variáveis categóricas, como o ambiente e a condição climática, foram codificadas de forma determinística por meio de *label encoding*, atribuindo um inteiro único e fixo a cada categoria, o que garante reprodutibilidade entre execuções. A coluna *Attack_Type* foi mapeada para rótulos numéricos conforme a classificação: 0 - Normal, 1 - DoS, 2 - *Fuzzing*, 3 - *RPM Spoofing*, 4 - *Gear Spoofing*. Essa etapa assegura que os algoritmos recebam dados consistentes e comparáveis entre os 45 conjuntos experimentais.

4.4 Algoritmos de classificação para detecção de intrusões

Os experimentos foram executados utilizando Python e empregando bibliotecas especializadas para processamento de dados e aprendizado de máquina conforme detalhado ao final desse capítulo. Foram implementados quatro algoritmos de classificação, sendo eles, *Decision Tree*, *Random Forest*, *XGBoost* e CNN. Esses algoritmos foram escolhidos por apresentarem diferentes níveis de complexidade e formas de aprendizado. Essa escolha permite comparar modelos mais simples com outros mais avançados na tarefa de detecção.

A *Decision Tree* foi utilizada como modelo de referência (*baseline*), por ser simples e fácil de interpretar. O *Random Forest* foi escolhido por ser mais robusto, pois combina várias árvores e reduz o *overfitting* (quando o modelo aprende bem os dados de treino, mas não generaliza para novos dados).

O *XGBoost* foi incluído por seu bom desempenho em dados estruturados e por lidar bem com desbalanceamento de classes (RAJAPAKSHA et al., 2023), além de ser eficiente em grandes volumes de dados.

Por fim, a CNN foi utilizada para capturar padrões sequenciais nos bytes das mensagens CAN, algo que modelos baseados em árvores não fazem diretamente. Para melhorar o treinamento, foram usadas técnicas como *batch normalization* (para estabilizar o aprendizado), *dropout* (para evitar dependência excessiva de padrões) e o otimizador

Adam, que ajusta automaticamente a taxa de aprendizado.

A utilização de algoritmos com diferentes características permite não apenas comparar seus desempenhos, mas também analisar como cada abordagem se comporta diante da variação dos cenários operacionais. Com isso, é possível identificar quais cenários representam maior dificuldade para a detecção de intrusões (QP2) e avaliar como a diversidade dos dados influencia a eficácia dos modelos (QP3).

A seleção de hiperparâmetros foi realizada de forma automática por meio do *framework* Optuna, que emprega otimização bayesiana com *Tree-structured Parzen Estimator* (TPE). O Optuna foi preferido em relação a abordagens de busca exaustiva (*grid search*) e aleatória (*random search*) porque o TPE modela a distribuição de probabilidade dos hiperparâmetros que levam a bons resultados, direcionando as tentativas seguintes para regiões promissoras do espaço de busca.

Essa abordagem reduz significativamente o número de avaliações necessárias para encontrar configurações competitivas, o que é especialmente relevante dado o custo computacional de treinar quatro modelos sobre 45 conjuntos de dados.

A métrica utilizada na otimização foi o *F1-score*, por penalizar igualmente erros nas classes minoritárias independentemente de seu suporte, evitando que a busca favoreça configurações que privilegiem a classe Normal em detrimento das classes de ataque.

Os espaços de busca, o número de *trials* e os hiperparâmetros finais selecionados para cada modelo são apresentados na Tabela 6 do Capítulo 6.

4.5 Métricas e protocolos de avaliação

O desempenho dos algoritmos foi avaliado utilizando métricas clássicas de classificação: precisão (*precision*), *recall* e *F1-score*. A precisão indica a proporção de instâncias corretamente classificadas entre todas as previstas como pertencentes a uma classe;

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}}$$

o *recall* mede a proporção de instâncias corretamente identificadas entre todas as instâncias reais de uma classe:

$$\text{Recall} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}}$$

e o *F1-score* representa a média harmônica entre precisão e *recall*, equilibrando os dois critérios em uma única métrica:

$$\text{F1-score} = \frac{2 \times \text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

O suporte (*support*) de uma classe indica o número de instâncias reais dessa classe no conjunto de teste. Ele não mede desempenho, mas pondera a contribuição de cada classe nas médias agregadas: classes com maior suporte influenciam mais o *F1-score* ponderado, refletindo sua maior representatividade nas condições reais de operação

Em cenários com desbalanceamento de classes, como é característico de dados de tráfego veicular, a acurácia global pode ser enganosa, uma vez que, um modelo que classifique tudo como normal atingiria acurácia elevada sem detectar nenhum ataque. Por esse motivo, o *F1-score* ponderado pelo suporte de cada classe foi adotado como métrica principal de comparação entre modelos, pois pondera a contribuição de cada classe pelo número de instâncias reais, atribuindo maior peso às classes com maior representação no conjunto de teste sem ignorar as classes minoritárias.

Para reduzir o risco de selecionar hiperparâmetros que funcionem bem apenas em uma partição específica dos dados, a validação cruzada com 5 partições (*k-fold* com $k = 5$) foi integrada ao processo de otimização conduzido pelo Optuna: a cada *trial*, o candidato a hiperparâmetro foi avaliado pela média do *F1-score* sobre os cinco folds antes de registrar o resultado. O valor $k = 5$ foi escolhido por representar um equilíbrio entre a estabilidade da estimativa e o custo computacional de re-treinamento, que cresce de forma linear com k .

Diferente de um *k-fold* aplicado sobre registros individuais, as partições foram formadas por grupos de cenários completos. Sendo assim, os 45 cenários foram divididos em 5 grupos de 9, de modo que cada grupo cobrisse proporções equivalentes de tipos de via e condições climáticas. A cada iteração do Optuna, 4 grupos foram usados para treinamento e 1 para validação interna.

Essa estratégia garante que a seleção de hiperparâmetros favorece configurações que generalizam para condições operacionais inéditas, refletindo o uso real de um IDS implantado em novos veículos ou ambientes. As métricas reportadas no Capítulo 6 referem-se à avaliação final sobre o conjunto de teste retido (20% dos dados, separado antes de qualquer otimização), distinto dos folds utilizados durante a busca de hiperparâmetros.

As métricas foram calculadas tanto para o *dataset* combinado quanto para agregações por clima e ambiente, utilizando médias ponderadas pelo suporte. Essa organização permite verificar não apenas o desempenho global, mas também a sensibilidade dos modelos às variações ambientais e operacionais. A análise dos resultados busca confirmar se as bases geradas são representativas e adequadas para uso em sistemas de detecção de intrusões, avaliando sua capacidade de refletir cenários realistas em veículos autônomos.

4.6 Ambiente e configuração dos experimentos

Os experimentos foram conduzidos sobre a base consolidada descrita no Capítulo 5. O detalhamento completo do protocolo encontra-se nos Capítulos 4 e 5. A Tabela 5 apresenta as principais ferramentas utilizadas durante os experimentos. Os hiperparâmetros finais obtidos pela busca do Optuna são apresentados na Tabela 6.

Tabela 5 – Configuração do ambiente computacional e ferramentas utilizadas

Item	Descrição
<i>Ferramentas e bibliotecas</i>	
CARLA Simulator	Simulação dos cenários veiculares
Python	Linguagem base de todos os scripts
CANTools	Conversão de telemetria para <i>frames</i> CAN
Pandas / NumPy	Manipulação e pré-processamento dos dados
Imbalanced-learn	Aplicação do SMOTE e <i>undersampling</i>
<i>Scikit-learn</i>	<i>Decision Tree</i> e <i>Random Forest</i>
<i>XGBoost</i>	Modelo de <i>gradient boosting</i>
<i>TensorFlow/Keras</i>	CNN com convolução 1D
Optuna	Otimização automática de hiperparâmetros (TPE)
Joblib	Paralelismo no treinamento e processamento em lote
<i>Configuração do servidor</i>	
Sistema Operacional	Ubuntu 22.04.5 LTS
Processador	Intel Xeon(R) E5-2699 2.30 GHz, 18 núcleos / 36 <i>threads</i>
Memória RAM	128 GB
<i>Graphics Processing Unit</i> (GPU)	NVIDIA GeForce RTX 3060, 8 GB VRAM (CUDA 11.x)

Tabela 6 – Hiperparâmetros finais selecionados pelo Optuna para cada algoritmo, obtidos por busca bayesiana com *Tree-structured Parzen Estimator* (TPE).

Modelo	Hiperparâmetro	Espaço de busca	Valor final
<i>Decision Tree</i>	Profundidade máxima	5 – 50	15
	Critério de divisão	Gini, Entropia	Entropia
	Mín. amostras por folha	1 – 20	4
<i>Random Forest</i>	Número de árvores	50 – 500	250
	Profundidade máxima	5 – 50	30
	Critério de divisão	Gini, Entropia	Gini
<i>XGBoost</i>	Taxa de aprendizado	0,01 – 0,3	0,12
	Número de estimadores	50 – 500	400
	Profundidade máxima	3 – 15	10
	<i>Subsampling</i>	0,5 – 1,0	0,85
CNN	Filtros convolucionais	32 – 256	128
	Taxa de <i>dropout</i>	0,1 – 0,5	0,25
	Taxa de aprendizado	0,0001 – 0,01	0,0008
	Épocas máximas	50 – 200	150

O desempenho foi medido por precisão, *recall* e *F1-score*. O *recall* é especialmente crítico em segurança veicular, pois um ataque não detectado pode causar dano físico real. O *F1-score* foi adotado como função objetivo na busca de hiperparâmetros, penalizando igualmente erros em todas as classes, respondendo à **QP3**.

5 DESENVOLVIMENTO E CARACTERIZAÇÃO DO DATASET DE INTRUSÃO CAN

A criação desta base de dados seguiu etapas bem definidas para atingir os resultados esperados. Um dos principais desafios foi converter a telemetria de um simulador em uma sequência de quadros binários compatíveis com o padrão CAN 2.0B, mantendo a qualidade dos dados. Essa etapa é importante, pois uma base mal construída pode fazer com que os modelos aprendam padrões errados em vez de reconhecer comportamentos reais. Este capítulo detalha cada fase dessa construção, descrevendo as decisões técnicas e as estratégias usadas para garantir a qualidade do conjunto de dados final.

5.1 Configuração do simulador CARLA e ambiente experimental

O simulador CARLA, baseado na *Unreal Engine*, foi utilizado como ambiente principal de geração de dados. A escolha se deu pela sua capacidade de reproduzir com fidelidade a dinâmica de um veículo, oferecer integração com sistemas de telemetria e permitir a injeção controlada de ataques e anomalias. As simulações foram executadas em três mapas, cada um representando um contexto de operação diferente.

O mapa urbano (Town03) tem alta densidade de tráfego, múltiplas interseções e sinais de trânsito, criando situações de condução mais complexas e com manobras frequentes. O mapa rural (Town07) é composto por estradas sinuosas, menor fluxo de tráfego e maior distância de visibilidade. O mapa de rodovia (Town04) representa cenários de alta velocidade e ultrapassagens. A Figura 6 mostra exemplos dos cenários simulados.

Para cada mapa, a densidade de tráfego foi mantida igual em todas as execuções, garantindo que as variações observadas fossem resultado apenas de mudanças nas condições climáticas ou nos tipos de ataque.

As manobras executadas foram escolhidas para cobrir comportamentos importantes para a análise de segurança do barramento CAN, como mudança de faixa, estacionamento, frenagem de emergência, seguimento de veículos e navegação em cruzamentos. Os níveis de autonomia variaram de condução assistida até autonomia total, evitando que a base representasse apenas o comportamento de um tipo específico de veículo autônomo. A variação do nível de autonomia altera substancialmente a origem e a natureza das decisões

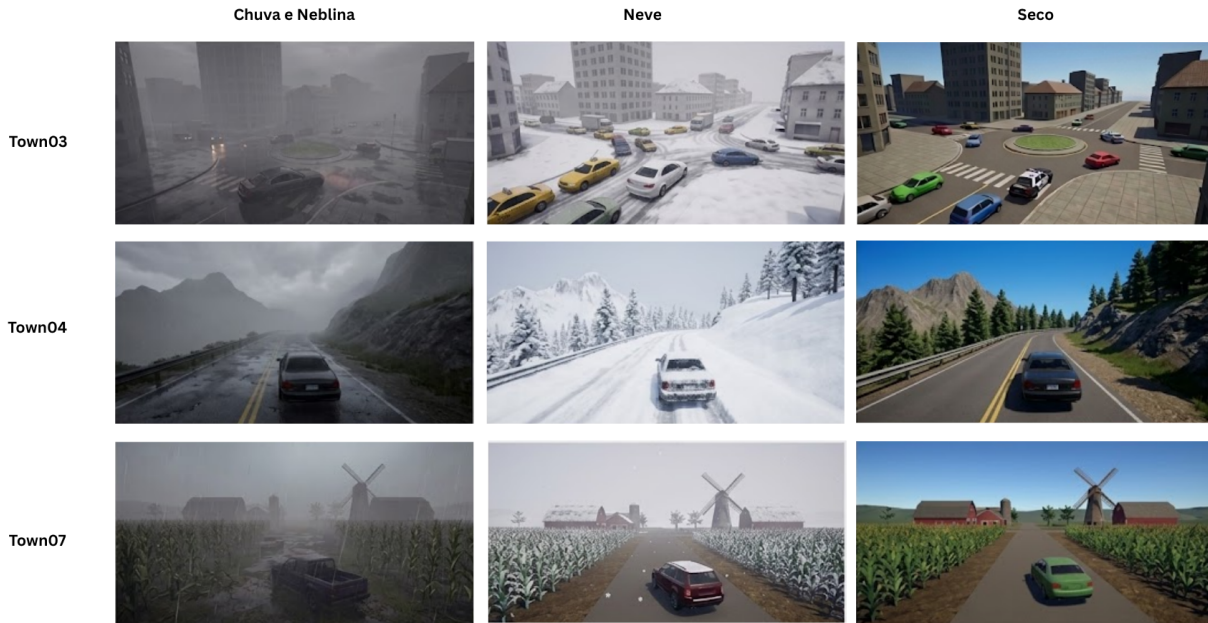


Figura 6 – Uma matriz 3x3 comparando as simulações em diferentes condições climáticas (Chuva/Neblina, Neve e Sol) em três cenários de simulação distintos: Urbano (Town03), Rodovia (Town04) e Rural (Town07).

que geram tráfego no barramento CAN. Em cenários de condução assistida, parte dos comandos de aceleração, frenagem e direção é originada por uma simulação de intervenção humana, com variabilidade e imprecisão características do comportamento de um condutor. Já em cenários de autonomia total, todos os comandos são gerados pelo CARLA, composto por controladores de trajetória e detectores de risco que avaliam continuamente a posição de pedestres, veículos, semáforos e sinalizações para decidir entre seguir a rota planejada ou executar frenagens de emergência (DOSOVITSKIY et al., 2017). Essa distinção é relevante porque a frequência, a regularidade e a previsibilidade dos sinais CAN gerados por decisões algorítmicas diferem das geradas por comportamento humano simulado, o que impacta diretamente a caracterização do tráfego normal usado como referência pelos modelos de detecção de intrusão. Por essa razão, a variação do nível de autonomia foi incorporada como parâmetro experimental, e não apenas como cenário de fundo, permitindo avaliar se os IDSs mantêm desempenho consistente independentemente da origem dos comandos enviados ao barramento.

O veículo foi equipado com sensores virtuais para coleta contínua de telemetria, incluindo velocidade, aceleração, rotação do motor e estado da transmissão. As simulações foram executadas em modo síncrono, que garante consistência entre o tempo do simulador e o processamento dos dados, pois o tempo só avança quando o cliente confirma que processou o quadro anterior.

5.2 Estratégia de construção de cenários

O conjunto de dados final é resultado da união de quarenta e cinco bases independentes, construídas a partir da combinação de três dimensões: ambiente (urbano, rural e rodovia), condição climática (seco, chuva com neblina e neve) e perfil de ataque (operação normal e mais quatro tipos de ataque). Cada combinação corresponde a um experimento independente, totalizando $3 \times 3 \times 5 = 45$ conjuntos distintos.

Cada arquivo CSV produzido representa uma combinação única dessas variáveis. Os nomes dos arquivos foram padronizados para facilitar a identificação, seguindo o formato `[identificador]_[tipo_via]_[condicao_clima]_[tipo_ataque].csv`. Por exemplo, o arquivo `attacked_urban_snow_dos.csv` contém amostras de uma trajetória urbana sob neve com injeção de ataque de negação de serviço. Essa divisão permite analisar subgrupos e investigar o efeito de cada ataque e de cada condição ambiental sobre o tráfego CAN de forma isolada.

Cada cenário foi executado por vinte minutos de tempo simulado, gerando cerca de 120.000 quadros CAN. O volume total de dados coletados no arquivo unificado `combined_can_dataset.csv` foi de aproximadamente 5,4 milhões de registros, reduzido para 3.636.091 registros após as etapas de pré-processamento, que incluíram a remoção de registros duplicados e quadros com campos corrompidos ou fora dos limites válidos do padrão CAN 2.0B.

5.3 Conversão de mensagens para o padrão CAN 2.0B

A biblioteca *CANTools* foi usada para montar as mensagens a partir de um arquivo DBC customizado, que define identificadores, formatos, fatores de escala e periodicidades compatíveis com o padrão. O mapeamento cobriu os sinais mais relevantes para a análise de segurança veicular.

A consistência das mensagens foi verificada comparando o valor decodificado com o original. Também foi verificada a coerência entre sinais relacionados: por exemplo, a aceleração registrada foi comparada com a variação de velocidade entre quadros seguintes. Esse procedimento garantiu que a conversão não introduzisse erros nos dados.

5.4 Influência do clima nos dados

As condições climáticas foram configuradas por meio dos parâmetros disponíveis na classe `carla.WeatherParameters` da API do simulador. Essa classe oferece atributos como `precipitation` (intensidade de chuva, de 0 a 100), `precipitation_deposits`

(acúmulo de água ou neve na pista, de 0 a 100), **wetness** (umidade da superfície, de 0 a 100) e **fog_density** (densidade de neblina, de 0 a 100), que alteram o comportamento da pista e do veículo. Essa abordagem garante que variações causadas por condições climáticas difíceis sejam incluídas na base, tornando o problema de detecção mais próximo de situações reais, já que o modelo precisa aprender a separar anomalias causadas pelo clima de anomalias causadas por ataques.

No cenário de neve, os parâmetros **precipitation_deposits** e **wetness** foram definidos com valores altos, simulando acúmulo de neve e gelo na pista. Isso reduz muito a aderência, o que muda bastante a dinâmica do veículo: o sistema de propulsão entra em regime de deslizamento, fazendo o RPM do motor subir sem que a velocidade do veículo aumente na mesma proporção. Esse fenômeno, conhecido como *spin-up* das rodas, é uma inconsistência física legítima que o detector precisa aprender a diferenciar de um ataque de *RPM Spoofing*. A relação entre variação de RPM e variação de velocidade pode ser de cinco a dez vezes maior do que em asfalto seco.

Em condições de chuva com neblina, os parâmetros **precipitation** e **fog_density** foram ajustados para valores moderados, representando chuva contínua com visibilidade reduzida. Nesse contexto, o comportamento do controlador autônomo muda: a velocidade tende a ser menor e o espaçamento entre veículos maior, para compensar a visibilidade e a aderência reduzidas. Essas mudanças alteram o padrão das mensagens CAN, gerando sequências com manobras mais suaves e espaçadas, o que torna o problema de detecção mais difícil.

Os cenários com clima seco serviram como referência, com os parâmetros de chuva e neblina zerados, representando asfalto limpo e seco em condições normais.

5.5 Implementação técnica dos ataques cibernéticos

Os ataques foram injetados diretamente na sequência de quadros CAN após a conversão da telemetria, simulando a ação de um nó malicioso conectado ao barramento. Essa abordagem representa situações reais em que um atacante obtém acesso ao barramento CAN por meio de portas de diagnóstico (OBD-II) ou por comprometimento de módulos periféricos.

O ataque de negação de serviço (DoS) foi implementado pela injeção de mensagens com o identificador de maior prioridade (*arbitration_id* = 0) em intervalos curtos, com oito bytes de *payload* com valor 255. Como o protocolo CAN dá prioridade às mensagens com IDs menores, um nó que transmite continuamente com o menor ID possível domina o barramento e impede que as ECUs legítimas enviem suas mensagens. O efeito esperado é o aumento de latência e eventual perda de mensagens críticas.

O ataque de manipulação de RPM (*RPM Spoofing*) consistiu em substituir mensagens legítimas por quadros com valores fisicamente impossíveis ou incompatíveis com a dinâmica do veículo. Foram usadas três estratégias: adição de um valor fixo ao RPM reportado; inversão da relação esperada, fazendo o RPM cair quando a velocidade sobe; e introdução de ruído de alta frequência na leitura de RPM. Cada estratégia testa uma capacidade diferente do detector: a primeira exige que o modelo perceba que RPM e velocidade não se correspondem; a segunda cria relações fisicamente erradas entre sinais relacionados; e a terceira verifica se o detector distingue ruído injetado de ruído natural do sensor.

O ataque de falsificação de marcha (*Gear Spoofing*) foi implementado pela alteração do byte que indica o estado da transmissão, gerando valores incompatíveis com a situação do veículo, como indicação de marcha ré com o veículo em alta velocidade, trocas de marcha em intervalos muito curtos ou indicação de estacionamento enquanto o veículo acelera. Esses padrões representam o comprometimento do módulo de controle de transmissão.

O ataque de *Fuzzing* injetou identificadores e *payloads* aleatórios no espaço de IDs válidos do CAN 2.0B, simulando tentativas de explorar falhas nos *parsers* das ECUs. Foram injetadas cerca de cinquenta mensagens por segundo durante a fase de ataque. O efeito é um aumento de variância no barramento que pode levar ECUs a entrar em modo de segurança.

A inserção dos quadros maliciosos foi guiada pela cadência original do barramento. Com o auxílio do recurso de localização por expressão regular de um editor de texto, identificou-se o intervalo de transmissão característico de cada CAN ID alvo, e os quadros injetados foram posicionados com o objetivo de preservar essa periodicidade, tornando a detecção dependente da análise semântica e não do formato ou da cadência das mensagens.

5.6 Unificação, rotulagem e estrutura final do *dataset*

Concluídas as etapas de geração, conversão e injeção de ataques, os quarenta e cinco conjuntos foram unidos no arquivo `combined_can_dataset.csv` por meio de um *script* de mesclagem. Durante esse processo, o *script* lê o nome de cada arquivo e extrai automaticamente o ambiente e a condição climática, adicionando as colunas `road_type` e `climate` a cada registro antes de unificá-los. Isso garante que cada amostra possa ser rastreada até seu contexto de origem sem que o pesquisador precise reconstruir essas informações manualmente. O processo incluiu ainda a verificação individual de cada arquivo quanto à integridade e à ordem dos registros.

5.6.1 Estrutura das colunas

A Tabela 7 descreve cada campo presente no arquivo `combined_can_dataset.csv`, com tipo de dado e valores possíveis. As colunas `road_type` e `climate` são adicionadas pelo *script* de mesclagem com base na origem de cada arquivo, permitindo análises por subgrupo diretamente no *dataset* consolidado.

Tabela 7 – Descrição das colunas presentes no arquivo `combined_can_dataset.csv`, com tipo de dado e valores possíveis para cada campo. As colunas `road_type` e `climate` são adicionadas pelo *script* de mesclagem; `flag` e `attack_type` operam em níveis distintos de rotulagem, conforme detalhado na Seção 5.

Coluna	Tipo	Valores possíveis	Descrição
<code>timestamp</code>	ponto flutuante	$\geq 0,0$	Instante de captura do <i>frame</i> (s)
<code>arbitration_id</code>	inteiro	0 – 2047 (CAN 2.0B 11 bits)	Identificador CAN do subsistema transmissor
<code>dlc</code>	inteiro	0 – 8	Comprimento do campo de dados (bytes)
<code>flag</code>	categórico	Normal, Malicious	Indica se o <i>frame</i> é legítimo ou malicioso
<code>attack_type</code>	categórico	Normal, DoS, Fuzzing, RPM_Spoofing, Gear_Spoofing	Contexto de ataque do cenário em que o <i>frame</i> foi capturado
<code>road_type</code>	categórico	urban, rural, highway	Ambiente do cenário de origem
<code>climate</code>	categórico	dry, rain_fog, snow	Condição climática do cenário de origem
<code>data_0 - data_7</code>	inteiro	0 – 255	Bytes 0 a 7 do <i>payload</i> CAN

Nota: Os nomes das colunas e seus respectivos valores foram mantidos em inglês para preservar a fidelidade aos identificadores originais do *dataset*.

5.6.2 Semântica dos campos `flag` e `attack_type`

É importante entender a diferença entre os campos `flag` e `attack_type`, pois eles atuam em níveis diferentes. O campo `flag` classifica cada quadro individualmente: `Normal` indica um quadro CAN legítimo gerado por uma ECU do veículo, enquanto `Malicious` indica um quadro injetado pelo atacante. O campo `attack_type` identifica a categoria do ataque em andamento, por exemplo `DoS`, `Fuzzing`, `RPM_Spoofing` ou `Gear_Spoofing`, independentemente de cada quadro individual ser legítimo ou malicioso.

Essa distinção é fundamental para o treinamento supervisionado. Durante a segunda metade de um experimento com *Gear Spoofing*, por exemplo, o barramento contém ao mesmo tempo quadros legítimos das ECUs (`flag = Normal`, `attack_type = Gear_Spoofing`) e quadros maliciosos injetados (`flag = Malicious`, `attack_type = Gear_Spoofing`).

5.6.3 Caracterização estatística e preparação final do dataset

O processo de coleta e simulação gerou 5.399.832 registros brutos, distribuídos entre os 45 cenários conforme a Tabela 8. A distribuição por categoria de ataque antes do balanceamento é apresentada na Tabela 9, mostrando a predominância de registros maliciosos (cerca de 81% do total) sobre os registros normais (cerca de 19%). Esse desequilíbrio é esperado em ambientes de teste de segurança, embora em veículos reais a proporção seja inversa.

Tabela 8 – Contagem de registros por combinação de ambiente e condição climática, antes do balanceamento.

Ambiente	Condição climática	Instâncias	Proporção (%)
Urbano	Seco	580.891	10,76
Urbano	Chuva + Neblina	599.428	11,10
Urbano	Neve	592.997	10,98
Rural	Seco	589.455	10,92
Rural	Chuva + Neblina	588.509	10,90
Rural	Neve	589.242	10,91
Rodovia	Seco	599.827	11,11
Rodovia	Chuva + Neblina	609.368	11,28
Rodovia	Neve	650.115	12,04
Total	–	5.399.832	100

Tabela 9 – Distribuição de registros por categoria de ataque antes do balanceamento (5.399.832 registros). O DoS concentra o maior volume em razão da injeção massiva de mensagens que caracteriza esse vetor, enquanto *RPM Spoofing* e *Gear Spoofing* apresentam os menores suportes.

Categoria	Flag	Instâncias	Proporção (%)
Normal	Normal	1.025.421	~19
DoS	Malicioso	2.116.707	~39
<i>Fuzzing</i>	Malicioso	1.194.054	~22
<i>RPM Spoofing</i>	Malicioso	553.955	~10
<i>Gear Spoofing</i>	Malicioso	509.695	~9
Total	–	5.399.832	100

A base foi dividida em 80% para treinamento (4.319.866 registros) e 20% para teste (1.079.966 registros). A divisão foi feita por cenário completo, e não de forma aleatória por registro, garantindo que dados de um mesmo cenário não apareçam nas duas partições

ao mesmo tempo. Esse cuidado evita vazamento de informação (*data leakage*) e produz estimativas de generalização mais confiáveis, o que é especialmente importante quando o objetivo é avaliar os modelos em cenários não vistos durante o treinamento. O conjunto de teste não passou por balanceamento, mantendo a distribuição original dos dados para que a avaliação dos modelos reflita condições reais de operação.

Sobre o conjunto de treinamento, aplicou-se uma combinação de redução das classes majoritárias (*undersampling*) com a técnica SMOTE nas classes minoritárias, igualando as cinco categorias em 511.225 instâncias cada, totalizando 2.556.125 registros de treinamento balanceados. A Tabela 10 resume essa divisão e a Tabela 11 apresenta a distribuição após o balanceamento.

Tabela 10 – Divisão treino e teste antes e após o balanceamento. A separação foi feita por cenário completo para evitar *data leakage*. O balanceamento foi aplicado exclusivamente ao conjunto de treinamento, preservando a distribuição original no conjunto de teste.

Partição	Pré-balanceamento	Pós-balanceamento
Treino	4.319.866	2.556.125
Teste	1.079.966	1.079.966
Total	5.399.832	3.636.091

Tabela 11 – Distribuição de registros por categoria após o balanceamento do conjunto de treinamento. A combinação de *undersampling* e *SMOTE* igualou as cinco categorias em 511.225 instâncias cada, totalizando 2.556.125 registros.

Categoria	Instâncias	Proporção (%)
Normal	511.225	20,0
DoS	511.225	20,0
<i>Fuzzing</i>	511.225	20,0
<i>RPM Spoofing</i>	511.225	20,0
<i>Gear Spoofing</i>	511.225	20,0
Total	2.556.125	100

Antes de disponibilizar o conjunto de dados, foram feitas verificações de qualidade, como análise de consistência em amostras aleatórias, inspeção das distribuições de cada coluna contínua e testes de separabilidade entre instâncias normais e anômalas. As verificações confirmaram que os ataques injetados produzem mudanças detectáveis nos dados,

o que justifica o uso de treinamento supervisionado. O *dataset* e o código-fonte completo de coleta e processamento foram disponibilizados para permitir que outros pesquisadores reproduzam e verifiquem os resultados de forma independente*.

* <<https://github.com/cart-pucminas/AV-IDS-Dataset>>

6 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados obtidos com os algoritmos de classificação aplicados aos 45 cenários simulados. Os resultados são organizados por condição climática, por ambiente e de forma geral. A análise busca entender como fatores ambientais e operacionais afetam o desempenho dos modelos e quais abordagens são mais eficazes para a detecção de intrusões em redes CAN, respondendo diretamente às questões de pesquisa levantadas no Capítulo 1.

6.1 Análise descritiva do *dataset*

A união dos 45 cenários gerou um arquivo `combined_can_dataset.csv` com uma base de 3.636.091 registros, cuja divisão entre partições é detalhada na Tabela 10. Do total de 5.399.832 registros originais, 4.319.866 (80%) foram usados para treinamento e 1.079.966 (20%) para o conjunto de teste, com a separação feita por cenário completo para evitar *data leakage*. O conjunto de treinamento foi balanceado com *undersampling* e SMOTE, resultando em 2.556.125 registros com distribuição igual entre as cinco categorias, conforme descrito no Capítulo 5. O conjunto de teste foi mantido com sua distribuição original para que a avaliação reflita condições reais de operação.

Do conjunto de teste de 1.079.966 registros, 440.935 foram separados como conjunto de validação durante a busca de hiperparâmetros com Optuna, mantendo-se isolados do processo de treinamento. Os 639.031 registros restantes formam o subconjunto de avaliação final, sobre o qual todas as métricas reportadas nas tabelas deste capítulo foram calculadas.

A Figura 7 mostra a distribuição de registros por tipo de ataque na base consolidada. O DoS representa a maior parte (2.116.707 registros), seguido por *Fuzzing* (1.194.054) e tráfego Normal (1.025.421). Essa proporção reflete a natureza do ataque DoS, que injeta muitas mensagens de alta prioridade, aumentando bastante o volume de tráfego no barramento durante os cenários de ataque.



Figura 7 – Distribuição de instâncias por categoria de ataque na base consolidada antes do balanceamento (5.399.832 registros). O *Denial of Service* (DoS) representa a maior parcela (2.116.707 registros, 39,2%) por causa da injeção massiva de mensagens que caracteriza esse ataque. O tráfego Normal corresponde a apenas 19,0% do total, refletindo o desequilíbrio entre operação legítima e maliciosa nos cenários de teste.

A diversidade geográfica e ambiental da base é mostrada na Figura 8, que apresenta a contagem de registros por combinação de ambiente e condição climática. A distribuição é bem equilibrada entre as nove células, com variação de apenas 12% entre o cenário com menor volume (Urbano & Seco, 580.891 registros) e o de maior volume (Rodovia & Neve, 650.115 registros). Essa diferença reflete a maior quantidade de mensagens CAN em rodovias durante condições de neve, onde o controlador autônomo aumenta a frequência de leitura dos sensores de tração.

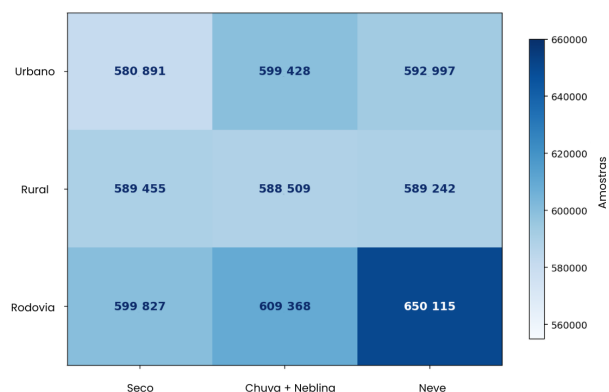


Figura 8 – Contagem de instâncias por combinação de ambiente e condição climática, mostrando a cobertura equilibrada dos 45 cenários. A variação entre a célula de menor volume (Urbano & Seco, 580.891 registros) e a de maior volume (Rodovia & Neve, 650.115 registros) é de apenas 12%.

6.2 Desempenho dos algoritmos de classificação

Os resultados foram organizados em três formas de análise: por condição climática, por ambiente e de forma global. Essa divisão permite tanto uma avaliação geral quanto a investigação de como fatores específicos afetam o desempenho de cada algoritmo.

6.2.1 Comparação geral de desempenho

Antes de discutir os resultados por contexto, é importante mostrar o desempenho geral de cada modelo. A Tabela 12 apresenta o *F1-score* médio ponderado pelo suporte de cada modelo por categoria de ataque, calculado como média das avaliações individuais dos 45 cenários.

Tabela 12 – *F1-score* médio $\pm$ desvio padrão por modelo e categoria de ataque, calculado sobre as avaliações individuais dos 45 cenários. O desvio padrão reflete a variação de desempenho entre cenários: classes de difícil detecção (*Spoofing*) e modelos mais simples apresentam maior dispersão. Os valores elevados de DoS ($\geq 0,98$) refletem avaliações isoladas; na avaliação conjunta do *dataset* consolidado (Tabela 25), os valores são menores em razão da competição simultânea entre todas as classes. A CNN obtém a maior média global (0,87) com a menor dispersão ($\pm 0,02$).

Categoria	Decision Tree	Random Forest	XGBoost	CNN
Normal	0,71 $\pm$ 0,06	0,79 $\pm$ 0,05	0,81 $\pm$ 0,04	0,84 $\pm$ 0,03
DoS	0,98 $\pm$ 0,01	0,99 $\pm$ 0,01	0,99 $\pm$ 0,01	0,99 $\pm$ 0,01
<i>Fuzzing</i>	0,79 $\pm$ 0,06	0,86 $\pm$ 0,05	0,90 $\pm$ 0,04	0,93 $\pm$ 0,03
RPM Spoofing	0,67 $\pm$ 0,08	0,78 $\pm$ 0,06	0,82 $\pm$ 0,05	0,85 $\pm$ 0,04
Gear Spoofing	0,64 $\pm$ 0,09	0,77 $\pm$ 0,07	0,80 $\pm$ 0,05	0,86 $\pm$ 0,04
Média ponderada	0,68 $\pm$ 0,05	0,80 $\pm$ 0,04	0,84 $\pm$ 0,03	0,87 $\pm$ 0,02

A Figura 9 complementa a tabela com um gráfico radar que permite comparar o perfil de cada algoritmo em todas as categorias de ataque ao mesmo tempo. A área da CNN é a maior em todas as dimensões, enquanto a *Decision Tree* apresenta as maiores quedas nos ataques de *Spoofing*.

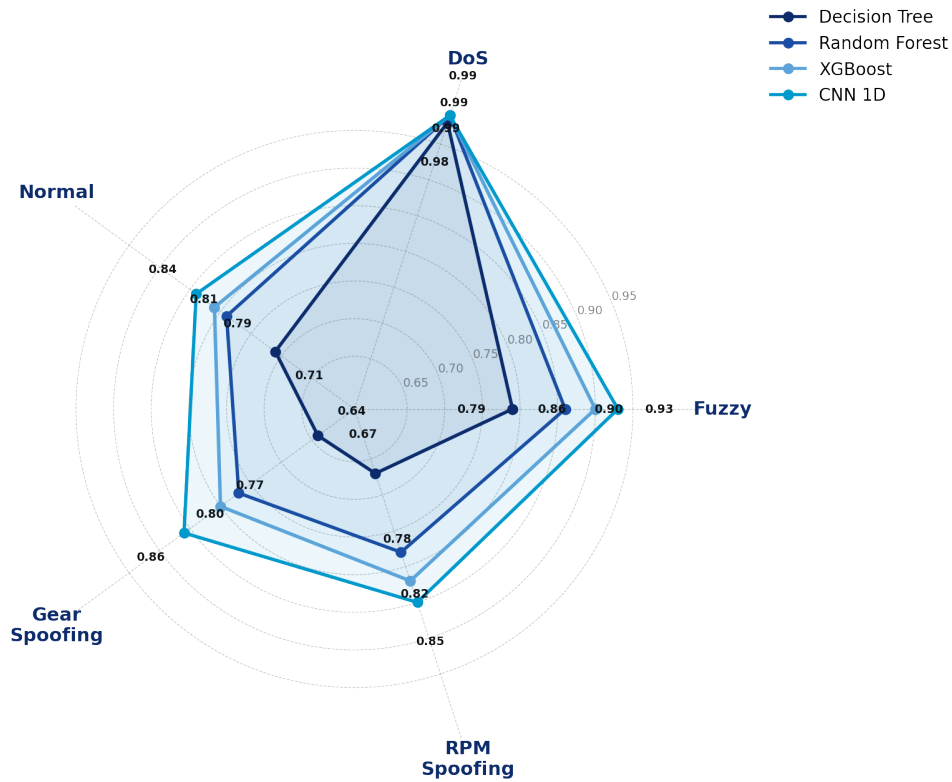


Figura 9 – $F1$ -score médio por algoritmo e categoria de ataque, consolidando todos os 45 cenários. Cada eixo representa uma categoria e a área preenchida reflete o desempenho geral de cada modelo. A CNN apresenta a maior cobertura em todas as dimensões, enquanto a *Decision Tree* apresenta as maiores quedas nas categorias de *Spoofing*, onde a relação entre sinais é essencial para a detecção.

A CNN superou todos os outros modelos em $F1$ -score médio ponderado (0,87), seguida pelo *XGBoost* (0,84), *Random Forest* (0,80) e *Decision Tree* (0,68). Essa ordem, apresentada na Tabela 12, se manteve em todos os cenários climáticos e ambientes, o que indica que a vantagem da CNN não vem de ajuste específico para um contexto, mas reflete diferenças reais na capacidade de cada algoritmo de identificar padrões nas sequências de bytes CAN, respondendo à **QP2** sobre quais características são mais relevantes para a detecção.

O desvio padrão reportado nas tabelas desta seção é calculado sobre as avaliações individuais dos 45 cenários, quantificando a estabilidade de cada modelo frente à variação de contexto. Dois padrões surgem diretamente dessa medida. Primeiro, modelos mais simples apresentam maior dispersão, refletindo alta sensibilidade às características do cenário. Segundo, classes de difícil detecção concentram maior variabilidade.

Um resultado que merece atenção é o desempenho na detecção de DoS. Nas avaliações individuais por cenário (Tabela 12), todos os modelos atingem $F1$ -score acima

de 0,98, pois a assinatura do ataque é tão clara que qualquer algoritmo a identifica com facilidade em condições ideais. Na avaliação do *dataset* consolidado (Tabela 25), onde todas as classes disputam ao mesmo tempo sobre a distribuição real do conjunto de teste, os valores de DoS caem para entre 0,70 (*Decision Tree*) e 0,87 (CNN): ainda os mais altos entre as categorias, mas mostrando que a competição entre classes e a variedade dos cenários reduz as métricas em relação às avaliações isoladas. Essa diferença entre os dois modos de avaliação é importante para ler corretamente as tabelas deste capítulo. Os ataques de *Spoofing* (RPM e Gear) são os mais difíceis em ambos os modos, pois exigem que o modelo identifique inconsistências entre sinais relacionados, e não apenas avalie valores individuais. Isso também se reflete no suporte de cada categoria: classes com menos exemplos, como *RPM Spoofing* e *Gear Spoofing*, têm menos peso nas médias e oferecem menos amostras de treinamento, dificultando a generalização do modelo.

A Tabela 13 detalha precisão e *recall* separados para os dois cenários mais críticos identificados na análise, onde a diferença entre variação legítima e ataque é mais sutil.

Tabela 13 – Precisão e *recall* detalhados para o melhor cenário combinado (Rodovia &Seco, *F1-score* médio CNN = 0,90) e o pior (Urbano &Neve, *F1-score* médio CNN = 0,81), evidenciando a amplitude real de variação do sistema quando ambiente e condição climática se combinam.

Cenário	Categoria	Precisão				Recall			
		DT	RF	XGB	CNN	DT	RF	XGB	CNN
Rodovia & Seco	DoS	0,76	0,86	0,90	0,93	0,74	0,85	0,88	0,91
	<i>Fuzzing</i>	0,75	0,85	0,89	0,92	0,73	0,84	0,87	0,91
	RPM Spoofing	0,75	0,85	0,89	0,92	0,73	0,83	0,87	0,90
	Gear Spoofing	0,74	0,84	0,88	0,92	0,72	0,83	0,86	0,90
	Normal	0,75	0,85	0,89	0,92	0,74	0,84	0,88	0,90
Urbano & Neve	DoS	0,62	0,71	0,78	0,83	0,60	0,70	0,76	0,81
	<i>Fuzzing</i>	0,61	0,71	0,77	0,82	0,60	0,70	0,76	0,81
	RPM Spoofing	0,61	0,71	0,77	0,82	0,59	0,69	0,75	0,80
	Gear Spoofing	0,60	0,70	0,76	0,81	0,59	0,69	0,75	0,79
	Normal	0,61	0,71	0,77	0,82	0,59	0,70	0,76	0,80

6.2.2 *Análise qualitativa dos padrões de erro*

Três tipos de equívoco de classificação surgem dos resultados, com implicações de segurança distintas. O mais grave é o falso negativo em ataques de *Spoofing*: um *frame* malicioso rotulado como Normal, permitindo que o ataque opere sem disparar qualquer alerta. Essa confusão ocorre porque RPM Spoofing e Gear Spoofing produzem valores de *payload* individualmente possíveis, ou seja, a inconsistência só se torna visível com o histórico de outros sinais. Modelos baseados em árvores, que analisam atributos de forma independente e sem memória temporal, são os mais suscetíveis a esse erro e apresentam as maiores quedas de F1 justamente nessas classes.

O segundo tipo é a confusão entre dois ataques, por exemplo, classificar RPM Spoofing como Gear Spoofing. A ameaça ainda é detectada e o sistema de defesa pode reagir, mas o diagnóstico do vetor de ataque é incorreto. Esse padrão é esperado entre RPM e Gear Spoofing, pois ambos modificam um campo de 16 bits dentro do *payload* e diferem principalmente no identificador CAN. Modelos que não aprendem a dependência conjunta entre identificador e conteúdo do *frame* são naturalmente mais propensos a essa troca.

O terceiro tipo, de menor severidade, é o falso positivo: uma mensagem legítima classificada como ataque. Ocorre com maior frequência em ambiente urbano sob neve, onde acelerações bruscas em baixa aderência e mudanças rápidas de marcha produzem padrões temporais semelhantes a assinaturas de intrusão, sobretudo para Gear Spoofing. A CNN apresenta menor taxa de falso positivo nessas condições por aprender a distinguir a coerência temporal das variações legítimas das inconsistências características de um *frame* manipulado.

6.2.3 *Resultados em diferentes condições climáticas*

A Tabela 14 apresenta o *F1-score* médio ponderado de cada modelo por condição climática, reunindo todos os ambientes e perfis de ataque dentro de cada clima.

Tabela 14 – $F1$ -score médio $\pm$ desvio padrão por modelo e condição climática. A neve apresenta maior dispersão que o clima seco, refletindo a interação variável entre a dinâmica de patinagem das rodas e os diferentes perfis de ataque em cada ambiente.

Condição climática	Decision Tree	Random Forest	XGBoost	CNN
Seco	0,72 $\pm$ 0,04	0,82 $\pm$ 0,03	0,86 $\pm$ 0,03	0,89 $\pm$ 0,02
Chuva + Neblina	0,69 $\pm$ 0,05	0,80 $\pm$ 0,04	0,84 $\pm$ 0,03	0,87 $\pm$ 0,03
Neve	0,63 $\pm$ 0,06	0,75 $\pm$ 0,05	0,79 $\pm$ 0,04	0,82 $\pm$ 0,03

Em clima seco, que serve como referência de operação ideal, a CNN alcança $F1$ -score médio de 0,89, com bom desempenho especialmente em *RPM Spoofing* (0,88) e *Gear Spoofing* (0,89). A *Decision Tree*, como o modelo mais simples, fica em torno de 0,72, estabelecendo a linha de base. O *XGBoost* ocupa uma posição intermediária com 0,86 (Tabela 15).

Tabela 15 – Resultados detalhados por categoria de ataque em condição climática seca, que serve como referência de operação ideal. A CNN alcança $F1$ -score médio de 0,89; a *Decision Tree* estabelece a linha de base em 0,72.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,74	0,72	0,73	82.124
	RPM Spoofing	0,73	0,71	0,72	21.579
	Gear Spoofing	0,72	0,70	0,71	19.693
	Fuzzing	0,73	0,71	0,72	46.300
	Normal	0,73	0,72	0,72	39.805
Random Forest	DoS	0,80	0,79	0,79	82.124
	RPM Spoofing	0,80	0,78	0,79	21.579
	Gear Spoofing	0,79	0,78	0,78	19.693
	Fuzzing	0,80	0,78	0,79	46.300
	Normal	0,80	0,80	0,80	39.805
XGBoost	DoS	0,88	0,86	0,87	82.124
	RPM Spoofing	0,87	0,86	0,86	21.579
	Gear Spoofing	0,86	0,85	0,85	19.693
	Fuzzing	0,87	0,85	0,86	46.300
	Normal	0,87	0,86	0,86	39.805
CNN	DoS	0,91	0,90	0,90	82.124
	RPM Spoofing	0,90	0,87	0,88	21.579
	Gear Spoofing	0,91	0,88	0,89	19.693
	Fuzzing	0,90	0,88	0,89	46.300
	Normal	0,91	0,89	0,90	39.805

No cenário de chuva com neblina, todos os modelos apresentam uma pequena queda

nas métricas, da ordem de 2 a 3 pontos percentuais em relação ao clima seco. Esse resultado é esperado: com visibilidade reduzida, o controlador autônomo adota velocidades menores e acelerações mais suaves, o que muda os padrões temporais das mensagens CAN. Mesmo com essa queda, a CNN mantém desempenho alto (0,87), mostrando uma resiliência que os modelos mais simples não conseguem replicar (Tabela 16).

Tabela 16 – Resultados detalhados por categoria de ataque em condição de chuva com neblina. Degradação de 2 a 3 pontos percentuais em relação ao clima seco para todos os modelos, com a CNN mantendo desempenho de 0,87.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,70	0,69	0,69	83.384
	RPM Spoofing	0,70	0,68	0,69	21.909
	Gear Spoofing	0,69	0,68	0,68	19.995
	Fuzzing	0,70	0,69	0,69	47.009
	Normal	0,70	0,69	0,69	40.416
Random Forest	DoS	0,78	0,77	0,77	83.384
	RPM Spoofing	0,78	0,76	0,77	21.909
	Gear Spoofing	0,77	0,76	0,76	19.995
	Fuzzing	0,78	0,77	0,77	47.009
	Normal	0,78	0,77	0,77	40.416
XGBoost	DoS	0,85	0,84	0,84	83.384
	RPM Spoofing	0,84	0,83	0,83	21.909
	Gear Spoofing	0,83	0,82	0,82	19.995
	Fuzzing	0,84	0,83	0,83	47.009
	Normal	0,84	0,83	0,83	40.416
CNN	DoS	0,88	0,87	0,87	83.384
	RPM Spoofing	0,89	0,87	0,88	21.909
	Gear Spoofing	0,88	0,87	0,87	19.995
	Fuzzing	0,88	0,86	0,87	47.009
	Normal	0,88	0,87	0,87	40.416

O cenário de neve é o mais difícil, com quedas em relação ao clima seco que variam conforme o modelo: 7 pontos percentuais para CNN, *Random Forest* e *XGBoost*, e 9 pontos percentuais para a *Decision Tree*. O baixo atrito provoca deslizamento de rodas e *spin-up* do motor sem aumento proporcional da velocidade do veículo, gerando uma diferença entre RPM e velocidade que, vista isoladamente, se parece com um ataque de *RPM Spoofing*. O principal problema não é deixar passar ataques reais, mas classificar operação normal em neve como anômala, ou seja, um aumento nos falsos positivos (Tabela 17).

Tabela 17 – Resultados detalhados por categoria de ataque em condição de neve. Cenário mais desafiador: queda de 7 a 9 pontos percentuais em relação ao seco, com aumento de falsos positivos em *RPM Spoofing* causado pelo *spin-up* legítimo das rodas em pista com baixa aderência.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,68	0,66	0,67	84.992
	RPM Spoofing	0,67	0,65	0,66	22.332
	Gear Spoofing	0,66	0,64	0,65	20.381
	Fuzzing	0,67	0,66	0,66	47.917
	Normal	0,67	0,66	0,66	41.195
Random Forest	DoS	0,75	0,73	0,74	84.992
	RPM Spoofing	0,73	0,72	0,72	22.332
	Gear Spoofing	0,73	0,71	0,72	20.381
	Fuzzing	0,74	0,73	0,73	47.917
	Normal	0,74	0,72	0,73	41.195
XGBoost	DoS	0,81	0,80	0,80	84.992
	RPM Spoofing	0,81	0,80	0,80	22.332
	Gear Spoofing	0,80	0,79	0,79	20.381
	Fuzzing	0,81	0,80	0,80	47.917
	Normal	0,81	0,80	0,80	41.195
CNN	DoS	0,86	0,84	0,85	84.992
	RPM Spoofing	0,82	0,84	0,83	22.332
	Gear Spoofing	0,83	0,81	0,82	20.381
	Fuzzing	0,85	0,82	0,83	47.917
	Normal	0,83	0,82	0,82	41.195

Sem informação sobre a condição climática, não é possível distinguir “RPM alto sem velocidade correspondente por causa do gelo” de “RPM alto sem velocidade correspondente por causa de um ataque”. A maior resistência da CNN nesse cenário confirma sua capacidade de identificar padrões complexos com múltiplas variáveis, mas mostra também a necessidade de IDSs que levem em conta o contexto ambiental, respondendo diretamente à **QP3**.

6.2.4 Resultados por ambiente

A Tabela 18 apresenta o *F1-score* médio por modelo e ambiente, reunindo todas as condições climáticas e perfis de ataque dentro de cada tipo de via.

Em rodovias, onde o tráfego é mais estável e previsível, a CNN alcança *F1-score* médio ponderado de 0,90 consolidando as três condições climáticas, com valor de 0,90 em clima seco, sendo o melhor resultado individual de toda a avaliação. A estabilidade dos sinais, com acelerações lentas e contínuas e baixa variação, facilita a detecção de ataques que introduzem mudanças bruscas ou inconsistências com os padrões físicos esperados.

Tabela 18 – *F1-score* médio $\pm$ desvio padrão por modelo e ambiente, calculado sobre os 15 cenários de cada grupo viário (3 condições climáticas $\times$ 5 perfis de ataque). O ambiente urbano apresenta maior dispersão em razão da alta variabilidade das manobras, que interage de forma diferente com cada perfil de ataque e condição climática.

Ambiente	Decision Tree	Random Forest	XGBoost	CNN
Rodovia	$0,73 \pm 0,04$	$0,83 \pm 0,03$	$0,87 \pm 0,03$	$0,90 \pm 0,02$
Rural	$0,70 \pm 0,04$	$0,81 \pm 0,03$	$0,85 \pm 0,03$	$0,88 \pm 0,02$
Urbano	$0,63 \pm 0,06$	$0,76 \pm 0,05$	$0,80 \pm 0,04$	$0,84 \pm 0,03$

O *XGBoost* mantém desempenho próximo (0,87 tanto em clima seco quanto como média das três condições) nesse ambiente, mostrando sua adequação para cenários com variação moderada (Tabela 19).

Tabela 19 – Resultados detalhados por categoria de ataque em rodovia. A estabilidade dos sinais em rodovias favorece todos os algoritmos: CNN (0,90) e *XGBoost* (0,87) apresentam os melhores resultados absolutos.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,75	0,73	0,74	86.245
	RPM Spoofing	0,74	0,72	0,73	22.661
	Gear Spoofing	0,73	0,71	0,72	20.681
	Fuzzing	0,74	0,72	0,73	48.623
	Normal	0,74	0,73	0,73	41.802
Random Forest	DoS	0,85	0,83	0,84	86.245
	RPM Spoofing	0,84	0,82	0,83	22.661
	Gear Spoofing	0,83	0,81	0,82	20.681
	Fuzzing	0,84	0,83	0,83	48.623
	Normal	0,84	0,82	0,83	41.802
XGBoost	DoS	0,89	0,88	0,88	86.245
	RPM Spoofing	0,88	0,86	0,87	22.661
	Gear Spoofing	0,87	0,86	0,86	20.681
	Fuzzing	0,88	0,86	0,87	48.623
	Normal	0,88	0,87	0,87	41.802
CNN	DoS	0,93	0,90	0,91	86.245
	RPM Spoofing	0,91	0,89	0,90	22.661
	Gear Spoofing	0,91	0,89	0,90	20.681
	Fuzzing	0,91	0,90	0,90	48.623
	Normal	0,91	0,89	0,90	41.802

Em estradas rurais, o desempenho continua alto para CNN e *XGBoost*: 0,88 e 0,85 como médias das três condições climáticas, com valores de 0,88 e 0,85 em clima seco. O ambiente rural tem características intermediárias: velocidades moderadas com boa visibilidade, mas com menos restrições de tráfego, o que permite manobras mais variadas. Essa combinação gera padrões de telemetria mais variáveis do que em rodovias, mas com estrutura regular o suficiente para que os algoritmos mais sofisticados distingam comportamento normal de comportamento comprometido (Tabela 20).

Tabela 20 – Resultados detalhados por categoria de ataque em estrada rural. Desempenho intermediário entre rodovia e urbano, com CNN em 0,88 e *XGBoost* em 0,85, refletindo a variabilidade moderada das manobras nesse ambiente.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,72	0,70	0,71	81.986
	RPM Spoofing	0,71	0,69	0,70	21.542
	Gear Spoofing	0,70	0,68	0,69	19.660
	Fuzzing	0,71	0,69	0,70	46.222
	Normal	0,71	0,70	0,70	39.739
Random Forest	DoS	0,82	0,81	0,81	81.986
	RPM Spoofing	0,82	0,80	0,81	21.542
	Gear Spoofing	0,81	0,80	0,80	19.660
	Fuzzing	0,82	0,81	0,81	46.222
	Normal	0,82	0,81	0,81	39.739
XGBoost	DoS	0,87	0,85	0,86	81.986
	RPM Spoofing	0,86	0,84	0,85	21.542
	Gear Spoofing	0,85	0,84	0,84	19.660
	Fuzzing	0,86	0,85	0,85	46.222
	Normal	0,86	0,85	0,85	39.739
CNN	DoS	0,90	0,88	0,89	81.986
	RPM Spoofing	0,89	0,87	0,88	21.542
	Gear Spoofing	0,89	0,87	0,88	19.660
	Fuzzing	0,89	0,87	0,88	46.222
	Normal	0,89	0,87	0,88	39.739

Ambientes urbanos apresentam o maior desafio para todos os algoritmos, com queda de 4 a 6 pontos percentuais em relação à rodovia. Acelerações frequentes, frenagens bruscas, mudanças de faixa e manobras de estacionamento criam uma grande variedade de padrões, com maior sobreposição entre classes. Nesse contexto, a CNN alcança *F1-score* de 0,84 em clima seco e como média das três condições, enquanto a diferença em relação ao *XGBoost* (0,80 em ambos) diminui. A *Decision Tree* sofre a maior queda (0,63 em clima seco), mostrando que modelos com menor capacidade de representação não conseguem lidar com a complexidade de separar variação natural de intrusão em ambiente urbano (Tabela 21).

Tabela 21 – Resultados detalhados por categoria de ataque em ambiente urbano. A *Decision Tree* sofre a degradação mais severa (0,63), evidenciando a insuficiência de modelos simples frente à alta sobreposição de classes nesse contexto.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,65	0,63	0,64	82.269
	RPM Spoofing	0,64	0,62	0,63	21.617
	Gear Spoofing	0,63	0,62	0,62	19.728
	Fuzzing	0,64	0,63	0,63	46.381
	Normal	0,64	0,62	0,63	39.875
Random Forest	DoS	0,77	0,76	0,76	82.269
	RPM Spoofing	0,77	0,75	0,76	21.617
	Gear Spoofing	0,76	0,75	0,75	19.728
	Fuzzing	0,77	0,76	0,76	46.381
	Normal	0,77	0,76	0,76	39.875
XGBoost	DoS	0,82	0,80	0,81	82.269
	RPM Spoofing	0,81	0,79	0,80	21.617
	Gear Spoofing	0,80	0,79	0,79	19.728
	Fuzzing	0,81	0,80	0,80	46.381
	Normal	0,81	0,80	0,80	39.875
CNN	DoS	0,86	0,84	0,85	82.269
	RPM Spoofing	0,85	0,83	0,84	21.617
	Gear Spoofing	0,84	0,82	0,83	19.728
	Fuzzing	0,85	0,84	0,84	46.381
	Normal	0,85	0,83	0,84	39.875

6.2.5 Efeito combinado de via e clima sobre a CNN

Para examinar como ambiente e condição climática se combinam, a Tabela 22 apresenta o *F1-score* médio da CNN para cada uma das nove combinações possíveis, calculado como média ponderada sobre todas as categorias de ataque dentro de cada célula via e clima.

Tabela 22 – *F1-score* médio da CNN para cada combinação de ambiente e condição climática. Dois gradientes simultâneos: queda de rodovia para urbano e de seco para neve. Amplitude total de 9 pontos percentuais entre o melhor (Rodovia & Seco, 0,90) e o pior cenário (Urbano & Neve, 0,81).

Ambiente	Seco	Chuva + Neblina	Neve
Urbano	0,84	0,83	0,81
Rural	0,88	0,86	0,84
Rodovia	0,90	0,88	0,85

A Figura 10 mostra esses valores visualmente, deixando claros dois gradientes: o desempenho cai de rodovia para urbano (eixo vertical) e de seco para neve (eixo horizontal). O pior resultado individual é Urbano & Neve (0,81) e o melhor é Rodovia & Seco (0,90), uma diferença de 9 pontos percentuais que mostra como a combinação de alta variação no trânsito e condição climática adversa piora os resultados de forma acumulada.

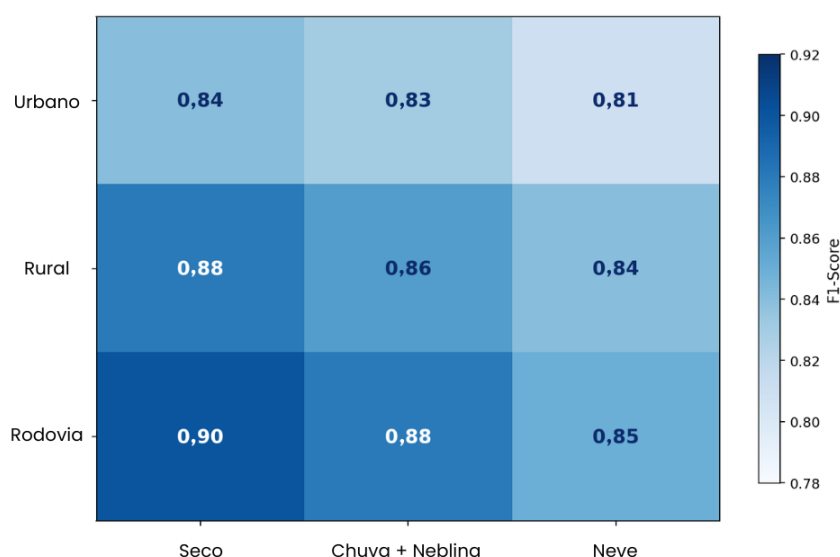


Figura 10 – *F1-score* médio da CNN para cada combinação de ambiente e condição climática. O cenário mais favorável é Rodovia & Seco (0,90) e o mais difícil é Urbano & Neve (0,81).

A análise separada por via e por clima, apresentada nas subseções anteriores, permite avaliar o efeito isolado de cada dimensão. Para ver como os dois fatores se combinam e se reforçam, a Tabela 23 apresenta o *F1-score* por categoria de ataque para cada uma das nove combinações de ambiente e condição climática, considerando todos os quatro algoritmos. A Tabela 24 foca na CNN, mostrando como cada categoria contribui para as diferenças observadas entre os cenários extremos.

Tabela 23 – *F1-score* por categoria de ataque para cada combinação de ambiente e condição climática, complementando as Tabelas 14 e 18 com a perspectiva cruzada. Dois gradientes simultâneos são observados: queda de Rodovia para Urbano e de Seco para Neve. A combinação Urbano & Neve representa o pior caso, com *Gear Spoofing* e *RPM Spoofing* como as categorias mais afetadas em todos os algoritmos. Valores em negrito referem-se à CNN.

Cenário	Algo.	DoS	<i>Fuzzing</i>	<i>RPM Sp.</i>	<i>Gear Sp.</i>	Normal
Rodovia & Seco	DT	0,77	0,76	0,76	0,75	0,76
	RF	0,85	0,84	0,84	0,83	0,85
	XGB	0,91	0,90	0,90	0,89	0,90
	CNN	0,93	0,92	0,91	0,93	0,93
Rodovia & Chuva + Neb.	DT	0,73	0,73	0,73	0,72	0,73
	RF	0,83	0,82	0,82	0,81	0,82
	XGB	0,88	0,86	0,86	0,85	0,86
	CNN	0,90	0,90	0,91	0,91	0,90
Rodovia & Neve	DT	0,71	0,70	0,70	0,69	0,70
	RF	0,80	0,78	0,77	0,77	0,78
	XGB	0,84	0,84	0,84	0,83	0,84
	CNN	0,88	0,86	0,86	0,85	0,85
Rural & Seco	DT	0,74	0,73	0,73	0,72	0,73
	RF	0,82	0,82	0,82	0,81	0,83
	XGB	0,89	0,88	0,88	0,86	0,88
	CNN	0,91	0,90	0,89	0,91	0,91
Rural & Chuva + Neb.	DT	0,70	0,70	0,70	0,69	0,70
	RF	0,80	0,80	0,80	0,79	0,80
	XGB	0,86	0,84	0,84	0,83	0,84
	CNN	0,88	0,88	0,89	0,89	0,88
Rural & Neve	DT	0,68	0,67	0,67	0,66	0,67
	RF	0,77	0,76	0,75	0,75	0,76
	XGB	0,82	0,82	0,82	0,81	0,82
	CNN	0,86	0,84	0,84	0,83	0,83
Urbano & Seco	DT	0,67	0,66	0,66	0,65	0,66
	RF	0,77	0,77	0,77	0,76	0,78
	XGB	0,84	0,83	0,83	0,82	0,83
	CNN	0,87	0,86	0,85	0,85	0,87
Urbano & Chuva + Neb.	DT	0,63	0,63	0,63	0,62	0,63
	RF	0,75	0,75	0,75	0,74	0,75
	XGB	0,81	0,80	0,80	0,79	0,80
	CNN	0,84	0,84	0,85	0,83	0,84
Urbano & Neve	DT	0,61	0,60	0,60	0,59	0,60
	RF	0,72	0,71	0,70	0,70	0,71
	XGB	0,77	0,77	0,77	0,76	0,77
	CNN	0,82	0,80	0,80	0,78	0,79

Tabela 24 – *F1-score* da CNN por categoria de ataque e combinação de ambiente e condição climática. Detalha como cada categoria contribui para os gradientes observados na Tabela 22. *RPM Spoofing* e *Gear Spoofing* apresentam as maiores quedas acumuladas (até 15 pp entre Rodovia & Seco e Urbano & Neve), enquanto DoS mantém a maior resiliência em todos os cenários.

Ambiente	Clima	DoS	<i>Fuzzing</i>	<i>RPM Sp.</i>	<i>Gear Sp.</i>	Normal
Rodovia	Seco	0,93	0,92	0,91	0,93	0,93
	Chuva + Neb.	0,90	0,90	0,91	0,91	0,90
	Neve	0,88	0,86	0,86	0,85	0,85
Rural	Seco	0,91	0,90	0,89	0,91	0,91
	Chuva + Neb.	0,88	0,88	0,89	0,89	0,88
	Neve	0,86	0,84	0,84	0,83	0,83
Urbano	Seco	0,87	0,86	0,85	0,85	0,87
	Chuva + Neb.	0,84	0,84	0,85	0,83	0,84
	Neve	0,82	0,80	0,80	0,78	0,79

Os resultados mostram que a queda combinada por via e clima é maior do que a observada em cada dimensão isolada. *Gear Spoofing* e *RPM Spoofing* são as categorias mais sensíveis à adversidade combinada: a CNN cai de 0,93 e 0,91 em Rodovia & Seco para 0,78 e 0,80 em Urbano & Neve, respectivamente, uma queda de 15 e 11 pontos percentuais — maior do que a soma das quedas individuais por via (6 pp) e por clima (7 pp), indicando que os dois fatores se reforçam de forma não linear.

6.2.6 Resultados consolidados do dataset combinado

Os resultados consolidados confirmam a ordem de desempenho observada nas análises por segmento (Tabela 25). A CNN obtém *F1-score* de 0,88 para *Gear Spoofing* e 0,86 para tráfego normal no *dataset* completo. O *XGBoost* alcança 0,82 e 0,81, respectivamente, uma diferença relevante, mas que ainda indica bom desempenho. A *Random Forest* apresenta resultados intermediários e a *Decision Tree* permanece como linha de base.

Tabela 25 – Relatório de classificação completo para o *dataset* consolidado (45 cenários, 639.031 registros do subconjunto de avaliação final, correspondente aos 1.079.966 registros de teste originais após reserva do conjunto de validação utilizado pelo Optuna). Apresenta precisão, *recall*, *F1-score* e suporte por categoria para os quatro algoritmos avaliados.

Algoritmo	Classe	Precisão	Recall	F1-Score	Suporte
Decision Tree	DoS	0,72	0,67	0,70	250.497
	RPM Spoofing	0,72	0,68	0,70	65.820
	Gear Spoofing	0,73	0,68	0,70	60.069
	Fuzzing	0,72	0,69	0,71	141.226
	Normal	0,70	0,68	0,69	121.419
Random Forest	DoS	0,78	0,75	0,77	250.497
	RPM Spoofing	0,79	0,74	0,77	65.820
	Gear Spoofing	0,77	0,73	0,75	60.069
	Fuzzing	0,78	0,75	0,77	141.226
	Normal	0,80	0,76	0,78	121.419
XGBoost	DoS	0,85	0,80	0,82	250.497
	RPM Spoofing	0,84	0,78	0,81	65.820
	Gear Spoofing	0,83	0,80	0,82	60.069
	Fuzzing	0,84	0,79	0,82	141.226
	Normal	0,82	0,80	0,81	121.419
CNN	DoS	0,88	0,85	0,87	250.497
	RPM Spoofing	0,87	0,86	0,87	65.820
	Gear Spoofing	0,90	0,86	0,88	60.069
	Fuzzing	0,89	0,85	0,87	141.226
	Normal	0,87	0,85	0,86	121.419

6.3 Comparação com o *Car-Hacking Dataset*

Para colocar os resultados em perspectiva, os mesmos algoritmos foram aplicados ao *Car-Hacking Dataset* (SEO; SONG; KIM, 2018), permitindo uma comparação entre as duas bases em condições experimentais iguais.

6.3.1 Desempenho no *Car-Hacking Dataset*

Os resultados obtidos no *Car-Hacking Dataset* são apresentados na Tabela 26. Embora os valores sejam razoáveis, a falta de variação ambiental na base limita o que os modelos conseguem aprender: todos os algoritmos apresentam desempenho parecido

entre si, com diferença de apenas 8 pontos percentuais entre o modelo mais simples e o mais sofisticado. Isso indica que a base não oferece desafio suficiente para separar o poder de cada abordagem.

Tabela 26 – Desempenho dos quatro algoritmos no *Car-Hacking Dataset* (SEO; SONG; KIM, 2018), sob protocolo experimental equivalente ao adotado para o *dataset* proposto.

Categoria	Decision Tree	Random Forest	XGBoost	CNN
DoS	0,85	0,88	0,90	0,87
<i>Fuzzing</i>	0,78	0,81	0,83	0,85
<i>RPM Spoofing</i>	0,65	0,69	0,72	0,75
<i>Gear Spoofing</i>	0,60	0,65	0,68	0,72
Média	0,72	0,76	0,78	0,80

6.3.2 Comparação com o *dataset* proposto

A Figura 11 mostra a comparação direta do *F1-score* da CNN entre as duas bases por categoria de ataque. O *dataset* proposto supera o *Car-Hacking* nas categorias que envolvem sinais físicos relacionados (*RPM Spoofing* e *Gear Spoofing*). No DoS, os dois *datasets* produzem o mesmo *F1-score* de 0,87 para a CNN quando avaliados pelo mesmo protocolo o *dataset* consolidado (Tabela 25) versus o *Car-Hacking* (Tabela 26). Esse resultado é esperado pois, a assinatura do ataque DoS é suficientemente clara para que qualquer base com volume adequado produza um detector igualmente eficaz nessa categoria.

A comparação entre a Tabela 26 e os resultados consolidados do *dataset* proposto (Tabela 12) mostra duas vantagens concretas da base com cenários heterogêneos.

A primeira é o desempenho absoluto superior. A CNN, melhor modelo em ambas as bases, alcança *F1-score* médio de 0,87 no *dataset* proposto contra 0,80 no *Car-Hacking*, diferença de 7 pontos percentuais que aumenta nos ataques de *Spoofing*. O treinamento em cenários variados expõe o modelo a situações legítimas de RPM e marcha que o *Car-Hacking*, coletado em condições estáticas, não contém. Um modelo treinado sem essa variedade tende a confundir operação normal em condições adversas com ataque, reduzindo sua utilidade prática.

A segunda é a maior diferença de desempenho entre algoritmos. No *Car-Hacking*, a diferença entre *Decision Tree* e CNN é de apenas 8 pontos percentuais (0,72 contra 0,80), o que dificulta justificar o custo computacional de modelos mais complexos. No *dataset*

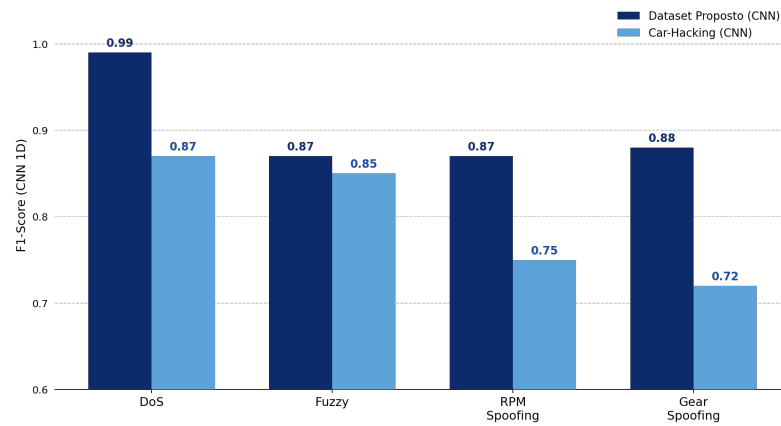


Figura 11 – Comparação do *F1-score* da CNN entre o *dataset* proposto e o *Car-Hacking Dataset* (SEO; SONG; KIM, 2018) por categoria de ataque. O *dataset* proposto supera o *Car-Hacking* nas categorias de *Spoofing* (RPM e Gear), onde a variedade de cenários expõe o modelo a variações legítimas ausentes na base de referência.

proposto, essa diferença sobe para 19 pontos percentuais (0,68 contra 0,87), deixando claro que a escolha do algoritmo tem impacto real no desempenho, o que é uma informação essencial para o projeto de IDSs embarcados com recursos limitados.

Vale destacar que as médias comparadas são ponderadas pelo suporte de cada base, que diferem entre si. Por exemplo, o *Car-Hacking* tem maior proporção de registros de DoS, o que eleva sua média global mesmo com desempenho inferior nos ataques de *Spoofing*.

Em resumo, o *dataset* proposto não apenas produz modelos com melhor desempenho, mas também oferece um ambiente de avaliação com maior capacidade de separar os algoritmos, revelando diferenças que bases homogêneas tendem a esconder.

6.4 Interpretação técnica e limitações observadas

6.4.1 Por qual motivo a CNN se destaca

A vantagem consistente da CNN está na sua capacidade de aprender como bytes vizinhos trabalham juntos para representar um único sinal físico. Modelos baseados em árvores tratam cada byte de forma independente, sem considerar a relação entre eles. A CNN, por outro lado, aprende que o byte 0 e o byte 1 de uma mensagem de RPM formam um inteiro de 16 bits, e que a combinação desses dois bytes representa um valor físico cujas variações devem ser coerentes com outros sinais do barramento.

Esse mecanismo permite à CNN detectar manipulações em que valores individuais dos bytes parecem normais isoladamente, mas cuja combinação produz um padrão fisi-

camente impossível. Um ataque que adiciona um desvio constante ao RPM pode fazer cada byte assumir valores plausíveis individualmente, e ainda assim a CNN identifica que a relação entre RPM e velocidade do veículo não corresponde ao que foi aprendido durante o treinamento, respondendo à **QP2** sobre as características mais relevantes para a detecção.

6.4.2 *Dificuldade na detecção de Gear Spoofing em ambientes urbanos*

O ataque de *Gear Spoofing* em ambientes urbanos apresenta o menor *F1-score* entre os cenários analisados. O motivo está na natureza do próprio trânsito urbano, que justifica mudanças frequentes de marcha em curtos intervalos: um veículo pode passar de marcha de condução para neutro em paradas e para ré em manobras de estacionamento em poucos segundos.

Quando o ataque injeta uma troca de marcha fora de contexto, como indicar ré enquanto o veículo avança em velocidade moderada, o detector pode classificar o evento como anômalo. Porém, em situações de congestionamento ou manobras complexas, há casos legítimos em que o estado reportado da transmissão pode estar atrasado em relação ao estado real. A diferença entre marcha falsificada e marcha com atraso legítimo de atualização é genuinamente difícil para um detector que opera apenas com os sinais do barramento CAN. Sem uma fonte externa de referência, como GPS confirmando a direção real do veículo, o detector enfrenta incerteza real nesse cenário específico.

6.4.3 *Influência climática e necessidade de contexto*

A análise no cenário de neve mostra que o principal problema não é deixar passar ataques reais — os ataques continuam sendo detectados com taxas razoáveis — mas sim o aumento de falsos positivos. Operações normais em neve são classificadas como anômalas com frequência muito maior do que em clima seco, especialmente na categoria de *RPM Spoofing*.

A explicação é direta: quando rodas patinam no gelo, o RPM sobe sem que a velocidade acompanhe proporcionalmente. Um detector treinado com dados de asfalto seco, onde essa relação é praticamente linear, interpreta essa diferença como sinal de ataque. O detector identifica corretamente a inconsistência entre sinais, mas não tem como reconhecer que essa inconsistência é esperada e legítima nas condições presentes.

Isso aponta uma limitação importante: IDSs baseados apenas nos padrões do barramento CAN, sem acesso a informações de contexto como condição climática ou estado de tração, precisam equilibrar sensibilidade em condições normais com tolerância a variações legítimas em condições extremas. Na prática, um sistema desse tipo precisaria receber

como entrada sinais adicionais de contexto, como temperatura, leituras de sensores de chuva ou aceleração lateral indicativa de perda de aderência, para ajustar os limiares de detecção de forma dinâmica, respondendo à **QP3**.

6.4.4 *Eficiência do XGBoost para ataques Fuzzing*

O *XGBoost* apresentou desempenho muito próximo ao da CNN na detecção de ataques *Fuzzing* (*F1-score* de 0,82), ficando apenas 5 pontos percentuais abaixo. Sua vantagem vem da natureza iterativa do *boosting*: a cada rodada, o algoritmo identifica os exemplos que os modelos anteriores erraram e constrói árvores focadas exatamente nesses casos. Para ataques *Fuzzing*, que por definição introduzem `arbitration_id` fora dos padrões aprendidos, o *boosting* cria naturalmente regras que identificam IDs raramente ou nunca vistos durante o treinamento.

Esse mecanismo torna o *XGBoost* uma boa opção para contextos em que clareza das regras de decisão e velocidade de resposta são importantes. Para aplicações embarcadas com recursos computacionais limitados, o *XGBoost* oferece um bom equilíbrio entre desempenho e custo computacional.

6.4.5 *Detectar versus interromper ataques DoS*

A detecção de DoS atingiu *F1-score* acima de 0,98 nos cenários individuais em condições ideais e 0,87 no *dataset* combinado para a CNN, o que poderia ser lido como o problema resolvido. No entanto, é importante separar detectar de interromper o ataque. A detecção é fácil porque o ataque quebra a estrutura temporal do barramento de forma imediata e clara. Já a interrupção continua sendo um problema de *hardware*: depois que o detector identifica o DoS, o barramento físico já está dominado pelo nó atacante, e sem intervenção em nível de *hardware* a detecção por si só não encerra o ataque. Esse achado reforça que a segurança de redes veiculares não pode depender apenas de algoritmos de detecção, mas exige proteção em múltiplas camadas.

6.5 Impacto da diversidade dos cenários no desempenho dos modelos

A decisão de incluir 45 cenários distintos, com combinações de três ambientes, três condições climáticas e cinco perfis de ataque, foi essencial para identificar limitações que teriam ficado escondidas em uma base menos variada. Se os experimentos tivessem se limitado a condições ideais, os modelos teriam apresentado desempenho acima de 0,90 em todas as métricas, o que não reflete os desafios reais encontrados em condições adversas.

A inclusão de ambientes urbanos com alta variação, condições climáticas extremas

e ataques de intensidade moderada revelou pontos fracos reais dos detectores. Esses pontos fracos não representam falhas dos modelos em si, mas refletem dificuldades que são intrínsecas ao problema: detectar intrusão quando a variação legítima e a anomalia maliciosa produzem padrões parecidos. A variedade dos dados, portanto, não apenas aumentou a confiabilidade das conclusões, mas permitiu identificar com precisão onde investimentos em defesas adicionais são mais necessários.

6.6 Limitações dos experimentos e caminhos futuros

Os resultados foram obtidos inteiramente em ambiente simulado. Embora o CARLA seja amplamente usado em pesquisa de veículos autônomos, a simulação simplifica aspectos de sistemas reais: sensores físicos apresentam padrões de falha diferentes dos da simulação, controladores reais têm características de implementação que afetam o tempo e as relações entre sinais de formas que o *software* não captura, e ataques reais podem explorar vulnerabilidades específicas de componentes não consideradas nos ataques sintéticos injetados.

Validar as conclusões aqui obtidas requer coleta de dados em veículos reais em ambientes controlados. Em particular, seria valioso verificar se a queda de desempenho em ambientes urbanos simulados se repete em condições reais; se a taxa de falsos positivos em neve simulada corresponde ao observado em neve real; e se os ataques injetados em simulação, quando aplicados contra *hardware* real, produzem padrões de tráfego CAN comparáveis aos da simulação.

Por fim, os ataques implementados representam um subconjunto das ameaças possíveis a redes CAN. Ataques mais sofisticados ou que modificam padrões de forma gradual para evitar os limiares de detecção não foram investigados. Avaliar detectores resistentes a ataques projetados especificamente para enganar os algoritmos aqui desenvolvidos representa um caminho importante para trabalhos futuros, assim como estender a metodologia para protocolos mais modernos como CAN-FD e Ethernet automotiva.

7 CONCLUSÕES

Este trabalho propôs a construção e avaliação de bases de dados com cenários heterogêneos para detecção de intrusão em redes *Controller Area Network* (CAN) de veículos autônomos. A pesquisa partiu da identificação de uma lacuna crítica na literatura, a ausência de *datasets* que representassem adequadamente a diversidade operacional de veículos autônomos, incluindo diferentes ambientes, condições climáticas adversas e múltiplos vetores de ataque. Para preencher essa lacuna, foram gerados 45 cenários simulados no CARLA, combinando três ambientes (urbano, rural e rodovia), três condições climáticas (seco, chuva com neblina e neve) e cinco perfis operacionais (normal e quatro categorias de ataque: DoS, *Fuzzing*, *RPM Spoofing* e *Gear Spoofing*).

O resultado dessa metodologia foi o arquivo `combined_can_dataset.csv`, com 3.636.091 registros estruturados no formato CAN 2.0B. O *dataset* adota uma estrutura dupla de rotulagem, onde o campo `flag` classifica cada *frame* individualmente como `Normal` ou `Malicious`, enquanto `attack_type` identifica a categoria específica do ataque em curso. Além disso, as colunas de metadados `road_type` e `climate` permitem análises por subgrupo sem reconstrução externa do contexto operacional. Essa estrutura diferencia o *dataset* das bases existentes, que em geral carecem de rotulagem contextual e são limitadas a condições específicas de operação, respondendo à **QP1**.

A avaliação dos algoritmos de classificação (*Decision Tree*, *Random Forest*, *XGBoost* e *CNN*) permite perceber que modelos mais sofisticados, em especial as redes neurais convolucionais, oferecem desempenho superior na identificação de mensagens maliciosas. Os resultados confirmam que técnicas de aprendizado de máquina são eficazes para detectar intrusões em redes CAN, mas também evidenciam que a qualidade e diversidade dos dados são determinantes para o sucesso dos modelos.

A *CNN* destacou-se por sua capacidade de capturar padrões sequenciais e sutis, decorrentes das relações estruturais entre bytes adjacentes e das correlações físicas entre sinais interdependentes, como RPM e velocidade, detectando manipulações em que valores individuais são plausíveis isoladamente, mas cuja combinação viola padrões físicos aprendidos durante o treinamento. Esse mecanismo permitiu à *CNN* alcançar *F1-score* superior a 0,80 em todos os cenários avaliados, com máximo de 0,90 em Rodovia & Seco, respondendo à **QP2**. O *XGBoost* apresentou desempenho consistente e robusto. Modelos mais simples, como *Decision Tree* e *Random Forest*, mostraram utilidade como linha de base, mas não alcançaram os mesmos níveis de generalização.

A análise por clima e ambiente revelou que fatores ambientais e operacionais influenciam diretamente o desempenho dos classificadores, com queda de métricas em ambientes urbanos e sob condições adversas, e aumento de falsos positivos em neve decorrente da semelhança entre variação legítima e anomalia maliciosa nesses contextos. Ainda assim, os modelos mais complexos mantiveram desempenho elevado, sugerindo maior resiliência frente ao ruído e à variabilidade, confirmando a **Hipótese 2**. Essa constatação reforça a importância de bases que contemplem múltiplos cenários, permitindo avaliar a robustez dos sistemas de detecção em diferentes contextos. O *F1-score* mostrou-se a métrica mais adequada como função objetivo por penalizar igualmente erros em todas as classes, enquanto o *recall* é a mais crítica sob a perspectiva de segurança veicular, uma vez que ataques não detectados representam risco físico real, respondendo à **QP3**.

A comparação direta com o *Car-Hacking Dataset*, conduzida com os mesmos algoritmos e configurações descritos no Capítulo 4, reforça a relevância da base proposta sob duas perspectivas complementares. Em termos de desempenho absoluto, a CNN alcançou *F1-score* médio de 0,87 na base heterogênea contra 0,81 no *Car-Hacking Dataset*, com a diferença se acentuando nos ataques de *Spoofing*, onde a variabilidade contextual dos cenários proporciona ao modelo uma compreensão mais rica da dinâmica veicular. Em termos de poder diagnóstico, a diferença entre o modelo mais simples e o mais sofisticado foi de 19 pontos percentuais na base proposta contra apenas 9 no *Car-Hacking Dataset*, evidenciando que bases homogêneas tendem a comprimir as métricas de todos os algoritmos próximas ao mesmo patamar, obscurecendo diferenças genuínas de capacidade de generalização que são determinantes para decisões de projeto de IDSs embarcados, confirmando a **Hipótese 1**.

Um desafio identificado foi a detecção de ataques de *Spoofing*, que imitam mensagens legítimas com pequenas alterações. Esses ataques exigem métodos mais sofisticados e possivelmente o uso de informações adicionais além do conteúdo das mensagens CAN, como correlações temporais ou estatísticas de fluxo. A diversidade dos cenários avaliados permitiu identificar com precisão essas vulnerabilidades específicas, em particular a ambiguidade do *Gear Spoofing* em congestionamento urbano e o aumento de falsos positivos de *RPM Spoofing* em neve, confirmando a **Hipótese 3** e evidenciando a necessidade de ampliar a exploração de características e experimentar abordagens híbridas que combinem aprendizado profundo com técnicas de análise temporal.

Em resumo, os resultados obtidos confirmam o objetivo central desta pesquisa: a criação de bases de dados representativas é fundamental para desenvolver sistemas de detecção de intrusão mais eficazes em veículos autônomos.

7.1 Trabalhos futuros

Os próximos passos desta pesquisa incluem a expansão das bases de dados, incorporando novos cenários de tráfego e condições operacionais ainda mais variadas. No que diz respeito às condições climáticas e ambientais, pretende-se incluir variações de luminosidade, como condução noturna e períodos de transição, que reduzam a luminosidade de forma total ou parcial. Além de temperaturas extremas e cenários com vias degradadas, condições que alteram os padrões de comunicação do barramento CAN de formas distintas das já contempladas. Serão também simulados diferentes regimes de densidade de tráfego, em especial o padrão *stop-and-go* em regiões de trânsito intenso, e ambientes com características de risco específicas, como zonas escolares, proximidades de hospitais e cruzamentos de alta complexidade, que introduzem velocidades reduzidas e maior densidade de pedestres.

Para validar as bases criadas, serão definidas métricas que avaliem a representatividade dos *datasets* e o equilíbrio entre classes de ataque. Esses *benchmarks* permitirão comparar diferentes abordagens de detecção quando aplicadas aos novos conjuntos, fornecendo parâmetros claros para avaliar sistemas de segurança veicular. Também será explorada a integração com dados coletados em veículos reais, de modo a verificar a aplicabilidade prática das soluções desenvolvidas em ambiente simulado.

A metodologia proposta permite a reprodução e expansão das bases por outros pesquisadores, facilitando a comparação de resultados e o avanço colaborativo na área. Essa abordagem, ao criar cenários diversos e realistas, é essencial para o desenvolvimento de sistemas de detecção de intrusão mais eficazes para veículos autônomos, contribuindo para a segurança de sistemas automotivos conectados.

Além disso, uma direção promissora é a utilização de ambientes integrados de simulação de *hardware* e *software*, como o SIMCenter*, para a coleta de dados em condições controladas com maior fidelidade ao comportamento elétrico e temporal de sistemas embarcados reais. Essa abordagem intermediária entre a simulação em *software* e a coleta em veículos completos permitiria validar e complementar os dados gerados no CARLA, reduzindo a lacuna entre os resultados obtidos em simulação e o desempenho esperado em implantações práticas de IDSs.

Por fim, pretende-se aprofundar a análise sistemática de vulnerabilidades específicas reveladas por bases heterogêneas, investigando de que forma cada combinação de ambiente, condição climática e perfil de ataque expõe limitações distintas nos detectores. Esse diagnóstico mais granular visa orientar o desenvolvimento de contramedidas direci-

* <<https://www.media.stellantis.com/br-pt/corporate-communications/press/fca-inaugura-primeiro-centro-de-simulacao-de-dinamica-veicular-da-america-latina>>

onadas, como mecanismos de resposta adaptativa e estratégias de mitigação específicas para cada vetor de ameaça identificado, contribuindo para a proteção mais eficaz das redes CAN de veículos autônomos contra os diferentes tipos de ataque catalogados.

7.1.1 *Produção decorrente da dissertação*

Este trabalho deu origem a uma publicação (AGUILAR; NOBRE; FREITAS, 2025) durante o seu desenvolvimento inicial. O artigo [†] foi apresentado no 57^o SBPO e aplicou os quatro algoritmos utilizados nesta dissertação ao *Car-Hacking Dataset*, analisando a detecção de intrusão em redes CAN. Os resultados serviram de referência para comparação e ajudaram a consolidar os experimentos antes de sua aplicação na base de dados desenvolvida neste trabalho.

Como parte da produção desta dissertação, há também as bases de dados geradas, que estão disponíveis no repositório GitHub.

- AV-IDS-Dataset, do inglês Autonomous Vehicle and Intrusion Detection System Dataset, encontra-se na seguinte URL: <https://github.com/cart-pucminas/AV-IDS-Dataset>

Além disso, os resultados obtidos nesta dissertação possibilitam novas publicações. A própria base de dados, os levantamentos sobre o impacto do clima e do ambiente na detecção de ataques e a comparação entre os algoritmos são contribuições com potencial para novas submissões, o que motiva a continuidade desta linha de pesquisa.

[†]AGUILAR, I. R.; NOBRE, C. N.; FREITAS, H. C. Analyzing Car-Hacking Dataset to Understand Cyberattacks and Intrusion Detection in Autonomous Vehicles. In: *57^o Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 2025, Gramado, RS. Anais do SBPO 2025. DOI: <<https://doi.org/10.59254/sbpo-2025-212171>>

REFERÊNCIAS

ABDULGANIYU, O. H.; TCHAKOUCHEV, T. A.; SAHEED, Y. K. A systematic literature review for network intrusion detection system (IDS). *INTERNATIONAL JOURNAL OF INFORMATION SECURITY*, Springer, v. 22, n. 5, p. 1125–1162, 2023.

AGUILAR, I. R.; NOBRE, C. N.; FREITAS, H. C. d. Analyzing Car-Hacking dataset to understand cyberattacks and intrusion detection in autonomous vehicles. In: *ANAIS DO LVII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL (SBPO)*. Gramado, RS: Galoá, 2025. Disponível em: <<https://doi.org/10.59254/sbpo-2025-21217>>. Acesso em: 08 maio 2026.

ANWAR, A. et al. Security assessment of in-vehicle communication protocols. *VEHICULAR COMMUNICATIONS*, Elsevier, v. 44, p. 100639, dec 2023. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S2214209623000694>>. Acesso em: 18 fev. 2026.

AVATEFIPOUR, O.; MALIK, H. State-of-the-art survey on in-vehicle network communication (CAN-bus) security and vulnerabilities. *INTERNATIONAL JOURNAL OF COMPUTER NETWORKS AND APPLICATIONS*, v. 5, n. 6, p. 1–17, 2018.

AWAAD, T. A. et al. Detecting cyber attacks in-vehicle diagnostics using an intelligent multistage framework. *SENSORS*, MDPI, v. 23, n. 18, p. 7941, 2023.

BANGUI, H.; GE, M.; BUHNOVA, B. A hybrid machine learning model for intrusion detection in VANET. *COMPUTING*, Springer, v. 104, n. 3, p. 503–531, 2022.

BOLAND, H. M. et al. An overview of CAN-bus development, utilization, and future potential in serial network messaging for off-road mobile equipment. *TECHNOLOGY IN AGRICULTURE*, IntechOpen, 2021.

BRASIL. LEI N. 9.503, DE 23 DE SETEMBRO DE 1997: INSTITUI O CÓDIGO DE TRÂNSITO BRASILEIRO. 1997. Diário Oficial da União. Disponível em: <https://www.planalto.gov.br/ccivil/_03/leis/l9503compilado.htm>. Acesso em: 18 fev. 2026.

BUSCEMI, A. et al. A survey on controller area network reverse engineering. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, IEEE, v. 25, n. 3, p. 1445–1481, 2023.

CHAWLA, N. V. et al. SMOTE: synthetic minority over-sampling technique. *JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH*, AI Access Foundation, v. 16, n. 1, p. 321–357, 2002.

CHO, K.-T.; SHIN, K. G. Fingerprinting electronic control units for vehicle intrusion detection. In: *25TH USENIX SECURITY SYMPOSIUM (USENIX SECURITY 16)*. Austin, TX, USA: USENIX Association, 2016. p. 911–927.

CNPq. PORTARIA CNPq Nº2.664, DE 6 DE MARÇO DE 2026: POLÍTICA DE INTEGRIDADE NA ATIVIDADE CIENTÍFICA DO CNPq. 2026. Diário Oficial da União, Seção 1, nº 47, 11 mar. 2026, p. 4. Acesso em: [inserir data]. Disponível em: <http://memoria2.cnpq.br/web/guest/view/-/journal/_content/56/_INSTANCE/_0oED/10157/23142775>.

DELAVARI, E.; KHANZADA, F. K.; KWON, J. A *COMPREHENSIVE REVIEW OF REINFORCEMENT LEARNING FOR AUTONOMOUS DRIVING IN THE CARLA SIMULATOR*. 2025. Preprint. Submetido em set. 2025. Disponível em: <<https://arxiv.org/abs/2509.08221>>. Acesso em: 29 jun. 2026.

DOSOVITSKIY, A. et al. CARLA: An open urban driving simulator. In: *PROCEEDINGS OF THE 1ST ANNUAL CONFERENCE ON ROBOT LEARNING*. [S.l.: s.n.], 2017. p. 1–16.

DUPONT, S. et al. ROAD: The ROad event Aware Detection framework for automotive Controller Area Network. In: *PROCEEDINGS OF THE 17TH EMBEDDED SECURITY IN CARS CONFERENCE (ESCAR)*. Cologne, Germany: [s.n.], 2019. p. 1–15.

FAWOLE, O. A.; RAWAT, D. B. Recent advances in 3D object detection for self-driving vehicles: a survey. *AI, MDPI*, v. 5, n. 3, p. 1255–1285, 2024.

HOSSAIN, M. D. et al. LSTM-based intrusion detection system for in-vehicle CAN bus communications. *IEEE ACCESS, IEEE*, v. 8, p. 185489–185502, 2020. Disponível em: <<https://ieeexplore.ieee.org/document/9216166>>. Acesso em: 18 fev. 2026.

IEHIRA, K.; INOUE, H.; ISHIDA, K. Spoofing attack using bus-off attacks against a specific ECU of the CAN bus. In: *IEEE. 2018 15TH IEEE ANNUAL CONSUMER COMMUNICATIONS & NETWORKING CONFERENCE (CCNC)*. Las Vegas, NV, USA, 2018. p. 1–4.

International Organization for Standardization. *ROAD VEHICLES — CONTROLLER AREA NETWORK (CAN) — PART 3: LOW-SPEED, FAULT-TOLERANT, MEDIUM-DEPENDENT INTERFACE*. Geneva, Switzerland, 2006.

KANG, M.-J.; KANG, J.-W. Intrusion detection system using deep neural network for in-vehicle network security. *PLOS ONE, Public Library of Science*, v. 11, n. 6, p. e0155781, 2016.

KANG, T. U. et al. Automated reverse engineering and attack for CAN using OBD-II. In: *IEEE. 2018 IEEE 88TH VEHICULAR TECHNOLOGY CONFERENCE (VTC-FALL)*. Chicago, IL, USA, 2018. p. 1–7.

KORIUM, M. S. et al. Intrusion detection system for cyberattacks in the internet of vehicles environment. *AD HOC NETWORKS, Elsevier*, v. 153, n. 1, p. 103330, 2024.

KOSCHER, K. et al. Experimental security analysis of a modern automobile. In: *IEEE. 2010 IEEE SYMPOSIUM ON SECURITY AND PRIVACY (SP)*. Oakland, CA, USA, 2010. p. 447–462.

LAMPE, B.; MENG, W. Intrusion detection in the automotive domain: A comprehensive review. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE*, v. 25, n. 4, p. 2356–2426, 2023.

LAMPE, B.; MENG, W. can-sleuth: Investigating and evaluating automotive intrusion detection datasets. In: PROCEEDINGS OF THE 2024 EUROPEAN INTERDISCIPLINARY CYBERSECURITY CONFERENCE (EICC '24). New York, NY, USA: Association for Computing Machinery, 2024. Disponível em: <<https://dl.acm.org/doi/10.1145/3655693.3655696>>. Acesso em: 18 fev. 2026.

LAMPE, B.; MENG, W. can-train-and-test: A curated CAN dataset for automotive intrusion detection. COMPUTERS & SECURITY, Elsevier, v. 140, p. 103777, 2024. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167404824000786>>. Acesso em: 18 fev. 2026.

LEE, H.; JEONG, S. H.; KIM, H. K. OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In: IEEE. 2017 15TH ANNUAL CONFERENCE ON PRIVACY, SECURITY AND TRUST (PST). Calgary, AB, Canada, 2017. p. 57–5709.

LI, Y. et al. *Choose your simulator wisely: A review on open-source simulators for autonomous driving*. IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE, v. 9, p. 4861–4876, 2024.

LIU, S. et al. Computer architectures for autonomous driving. COMPUTER, IEEE, v. 50, n. 8, p. 18–25, 2017.

LOKMAN, S.-F.; OTHMAN, A. T.; ABU-BAKAR, M.-H. Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. EURASIP JOURNAL ON WIRELESS COMMUNICATIONS AND NETWORKING, Springer, v. 2019, n. 1, p. 1–17, 2019.

LONGARI, S. et al. CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network. IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE, v. 17, n. 4, p. 2267–2278, 2020.

MARCHETTI, M.; STABILI, D. READ: Reverse engineering of automotive data frames. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE, v. 14, n. 4, p. 1083–1097, 2019.

MASEER, Z. K. et al. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE ACCESS, IEEE, v. 9, p. 22351–22370, 2021.

MILLER, C.; VALASEK, C. A survey of remote automotive attack surfaces. BLACK HAT USA, v. 2014, p. 1–20, 2014. Disponível em: <<https://illmatix.com/Remote/%20Car/%20Hacking.pdf>>. Acesso em: 18 fev. 2026.

MILLER, C.; VALASEK, C. REMOTE EXPLOITATION OF AN UNALTERED PASSENGER VEHICLE. 2015. Black Hat USA 2015. Whitepaper técnico. Apresenta acesso remoto ao barramento CAN de um Jeep Cherokee 2014 por meio do módulo de conectividade celular UConnect, com controle físico de motor, transmissão e freios. Disponível em: <https://www.ioactive.com/wp-content/uploads/pdfs/IOActive_Remote_Car_Hacking.pdf>. Acesso em: 18 fev. 2026.

- MOQVIST, E. *CANTOOLS: CAN BUS TOOLS*. 2024. Software. Biblioteca Python de código aberto para codificação, decodificação e análise de mensagens CAN a partir de arquivos DBC, KCD, SYM, ARXML e CDD. Disponível em: <<https://github.com/cantools/cantools>>. Acesso em: 29 jun. 2026.
- MÜTER, M.; ASAJ, N. Entropy-based anomaly detection for in-vehicle networks. In: IEEE. 2011 IEEE INTELLIGENT VEHICLES SYMPOSIUM (IV). Baden-Baden, Germany, 2011. p. 1110–1115.
- OTOUM, Y.; NAYAK, A. AS-IDS: Anomaly and signature based IDS for the internet of things. *JOURNAL OF NETWORK AND SYSTEMS MANAGEMENT*, Springer, v. 29, n. 3, p. 23, 2021.
- RAJAPAKSHA, S. et al. AI-based intrusion detection systems for in-vehicle networks: A survey. *ACM COMPUTING SURVEYS*, ACM, v. 55, n. 11, p. 1–40, feb 2023. Disponível em: <<https://dl.acm.org/doi/10.1145/3570954>>. Acesso em: 18 fev. 2026.
- SANABOINA, S. V. S. V. P.; NAIK, M. C.; RAJIV, K. Examining the impact of artificial intelligence methods on intrusion detection with the NSL-KDD dataset. In: IEEE. 2023 FIRST INTERNATIONAL CONFERENCE ON CYBER PHYSICAL SYSTEMS, POWER ELECTRONICS AND ELECTRIC VEHICLES (ICPEEV). Bengaluru, India, 2023. p. 1–7.
- SEO, E.; SONG, H. M.; KIM, H. K. GIDS: GAN based intrusion detection system for in-vehicle network. In: IEEE. 2018 16TH ANNUAL CONFERENCE ON PRIVACY, SECURITY AND TRUST (PST). Belfast, UK, 2018. p. 1–6.
- SILVA, E. et al. A systematic mapping of autonomous vehicle prototypes: Trends and opportunities. *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, IEEE, v. 9, n. 1, p. 1–27, 2024.
- SINGH, S. CRITICAL REASONS FOR CRASHES INVESTIGATED IN THE NATIONAL MOTOR VEHICLE CRASH CAUSATION SURVEY. Washington, DC, 2015. Disponível em: <<https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115>>. Acesso em: 18 fev. 2026.
- SkyQuestt. AUTONOMOUS VEHICLES MARKET SIZE, SHARE & INDUSTRY ANALYSIS, BY COMPONENT, BY LEVEL OF AUTOMATION, BY APPLICATION, AND BY REGION — FORECAST 2026–2033. 2025. Relatório de mercado. Valor base: USD 69,26 bi (2024); projeção: USD 719,35 bi (2033); CAGR: 29,7% (2026–2033). Disponível em: <<https://www.skyquestt.com/report/autonomous-vehicles-market>>. Acesso em: 18 fev. 2026.
- SONG, H. M.; KIM, H. R.; KIM, H. K. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In: IEEE. 2016 INTERNATIONAL CONFERENCE ON INFORMATION NETWORKING (ICOIN). Kota Kinabalu, Malaysia, 2016. p. 63–68.
- TAHIR, N. U. A. et al. Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches. *ALGORITHMS*, MDPI, v. 17, n. 3, p. 103, 2024.

TAYLOR, A.; LEBLANC, S.; JAPKOWICZ, N. Anomaly detection in automobile control network data with long short-term memory networks. In: IEEE. 2016 IEEE INTERNATIONAL CONFERENCE ON DATA SCIENCE AND ADVANCED ANALYTICS (DSAA). Montreal, QC, Canada, 2016. p. 130–139.

International Organization for Standardization. ROAD VEHICLES — CONTROLLER AREA NETWORK (CAN) — PART 2: HIGH-SPEED MEDIUM ACCESS UNIT. Geneva, Switzerland, 2003.

WANG, Q.; SAWHNEY, S. VeCure: A practical security framework to protect the CAN bus of vehicles. INTERNET OF THINGS, Elsevier, v. 14, p. 100303, 2020.

Waymo LLC. WAYMO OPEN DATASET. 2019. Base de dados de percepção ambiental composta por nuvens de pontos LiDAR, imagens de câmeras e trajetórias de objetos, destinada ao desenvolvimento de algoritmos de detecção, rastreamento e predição de movimento em veículos autônomos. Não contém dados de barramento CAN. Disponível em: <<https://waymo.com/open>>. Acesso em: 18 fev. 2026.

ZHU, M. et al. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. TRANSPORTATION RESEARCH PART C: EMERGING TECHNOLOGIES, Elsevier, v. 117, p. 102662, 2020.