Modelos Interpretáveis com Inteligência Artificial Explicável (XAI) na Detecção de Intrusões em Redes Intra-Veiculares Controller Area Network (CAN)

Felipe N. Dresch[1], **Felipe H. Scherer**[1], **Silvio E. Quincozes**[1],[2]
Diego Kreutz[1]

[1]LEA, PPGES, Universidade Federal do Pampa (UNIPAMPA) – Alegrete, Brasil.

[2]PPGCO, Universidade Federal de Uberlândia (UFU) – Uberlândia, Brasil

{felipedresch, felipescherer}.aluno@unipampa.edu.br

{silvioquincozes, diegokreutz}@unipampa.edu.br

Abstract. *In-vehicle networks that use the Controller Area Network (CAN) protocol are vulnerable to attacks such as fuzzing, fabrication, DoS, spoofing, replay, message injection, and fault injection. Existing studies typically address this problem through Intrusion Detection Systems (IDSs). However, these IDSs often lack explainability, compromising their reliability and interpretability, especially in CAN networks where information patterns are varied. This study explores the explainability of IDSs in CAN networks using the X-CANIDS dataset, which contains real vehicle data. The SHAP library was employed to provide model explainability, revealing relationships between CAN messages and attacker behavior, contributing to the interpretation of IDS decisions.*

Resumo. *Redes intra-veiculares que utilizam o protocolo Controller Area Network (CAN) são vulneráveis a ataques como fuzzing, fabricação, DoS, spoofing, replay, injeção de mensagens e injeção de falhas. Estudos existentes tipicamente abordam esse problema por meio de Sistemas de Detecção de Intrusões (IDSs). Contudo, esses IDSs frequentemente carecem de explicabilidade, o que compromete sua confiabilidade e interpretabilidade, especialmente em redes CAN, onde os padrões de comunicação são variados. Este estudo investiga a explicabilidade dos IDSs em redes CAN, utilizando o conjunto de dados X-CANIDS, que contém dados reais de veículos. A biblioteca SHAP foi empregada para fornecer explicabilidade ao modelo, revelando as relações entre mensagens CAN e o comportamento dos atacantes, contribuindo para uma melhor interpretação das decisões do IDS.*

1. Introdução

A segurança de redes intra-veiculares do tipo *Controller Area Network* (CAN) é um campo em constante evolução e alvo recorrente de críticas devido à ausência de mecanismos robustos para a garantia da segurança cibernética [Buscemi et al. 2023]. Os ataques em redes CAN ocorrem, em parte, pela fragilidade dos protocolos de segurança, que se deve aos limitados recursos computacionais disponíveis, o que inviabiliza a adoção de medidas de segurança convencionais tipicamente utilizadas em outras redes. Como resultado, as redes CAN têm se tornado um alvo atrativo para invasores [Jeong et al. 2024].

É importante destacar também que as transmissões realizadas nessa rede demandam baixa latência, já que informações e eventos devem ocorrer em tempo real para garantir o bom funcionamento do veículo. Para realizar transmissões dentro da rede, utilizam-se Unidades de Controle Eletrônico, do inglês *Electronic Control Unit* (ECU), que são responsáveis por receber os sinais de outros dispositivos e emiti-los na rede. A falta de medidas efetivas de segurança nesse tipo de protocolo resulta em significativas vulnerabilidades, fato exemplificado pela ausência de autenticação de mensagens. Isso significa que o destinatário de um pacote não consegue verificar a sua legitimidade e, com isso, ECUs comprometidas podem ser utilizadas por atacantes para enviar pacotes falsos com a intenção de comprometer o sistema [Lokman et al. 2019].

Com o objetivo de mitigar possíveis incidentes de segurança, trabalhos recentes têm avançado significativamente na utilização de Sistemas de Detecção de Intrusões (IDSs) baseados em Inteligência Artificial (IA), os quais têm demonstrado resultados promissores na detecção de intrusões [Jeong et al. 2024, Moulahi et al. 2021, Quincozes et al. 2023]. Apesar de sua eficácia, o uso de mecanismos baseados em IA frequentemente levanta dúvidas quanto à confiabilidade dos sistemas, uma vez que os motivos por trás de suas decisões são obscurecidos devido à complexidade dos algoritmos.

Além disso, esse tipo de sistema não permite inferir muito a respeito dos ataques reconhecidos, nem sobre eventuais falhas na detecção. Nesse contexto, a aplicação de ferramentas de Inteligência Artificial Explicável (XAI) torna-se um recurso essencial para entender os mecanismos de ataque e identificar falhas nos algoritmos implementados em IDSs [Gunning et al. 2019, Swetha et al. 2023, Jeong et al. 2024, Quincozes et al. 2024].

Diante das limitações da literatura em apresentar abordagens que considerem a explicabilidade de IDSs aplicados às redes CAN e levando em conta as vulnerabilidades inerentes a esse tipo de rede, este trabalho propõe o uso de técnicas de XAI para elucidar de forma clara e explicativa a relevância das características — também chamadas de *features*, compostas por informações contidas nos quadros transmitidos pela rede CAN — na detecção de intrusões. Ademais, a relação dessas características com o modo de operação dos atacantes é mapeada, de modo a fornecer uma interpretação das ações maliciosas executadas por eles. Como resultado, este trabalho contribui para a evolução dos algoritmos utilizados em IDSs aplicados às redes CAN, avançando na direção da adoção extensiva e compreensiva de IDSs Explicáveis, ou *eXplainable* IDSs (X-IDS).

Em particular, este trabalho se destaca por estudar o conjunto de dados recente denominado X-CAN-IDS [Jeong et al. 2024], o qual contém dados reais e contempla ataques como *fuzzing* e fabricação (*fabrication*). Para identificar e classificar esses ataques, emprega-se o XGBoost [Dhaliwal et al. 2018] (Seção 5), um algoritmo classificador popular e de comprovada eficiência na detecção de intrusões. Para obter a explicabilidade do modelo gerado pelo XGBoost, emprega-se o uso da biblioteca SHAP (*SHapley Additive exPlanations*), como detalhado na Seção 6. No entanto, a verdadeira explicabilidade dos dados só pode ser alcançada com conhecimento técnico detalhado do domínio, algo que este trabalho busca aprimorar através de engenharia reversa e da análise de documentação específica, como pode ser observado nas discussões da Seção 6.2.

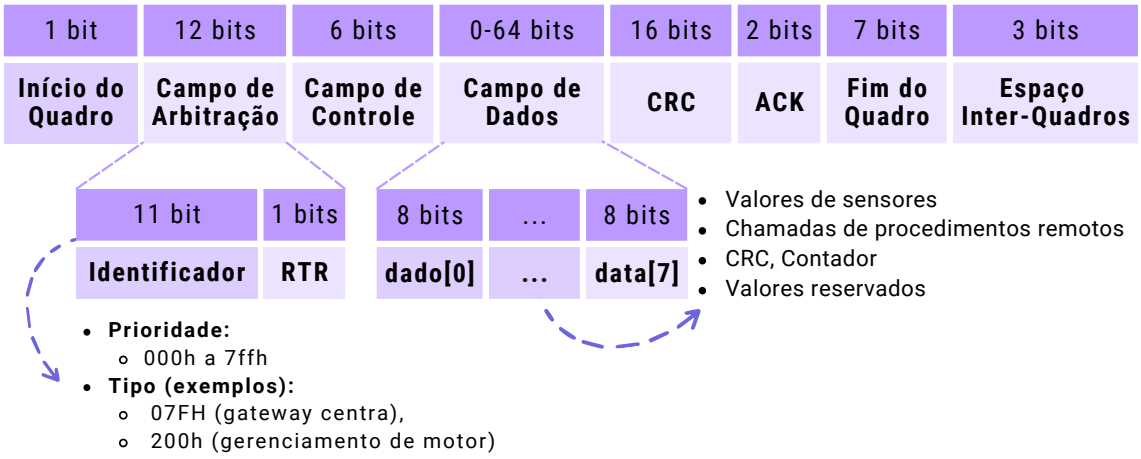

Figura 1. Quadro CAN (autoria própria, baseado em [Jeong et al. 2024]).

2. Redes CAN e Modelos de Ataques

As redes *Controller Area Network* (CAN) são assim chamadas devido ao uso do protocolo CAN. Em particular, este é um protocolo de comunicação utilizado para permitir a troca de informações entre as várias ECUs dentro de um veículo. Esses componentes possuem uma série de sensores que são responsáveis por coletar dados diversos, tais como a temperatura do motor, velocidade e torque do veículo [Jeong et al. 2024].

Cada ECU reúne os sinais coletados por seus sensores acoplados e transmite esses sinais na rede através de um quadro CAN, conforme ilustra a Figura 1. O quadro CAN é composto de diversos atributos, conforme é possível visualizar na figura. Dentre esses, destacam-se: i) o "Identificador de Arbitração" – *Arbitration ID* (AID) – que é o ID da mensagem cujo valor é utilizado como parâmetro para definir a prioridade de uma mensagem, sendo valores mais baixos prioritários; e ii) o "Campo de Dados" (do inglês, *Data Payload*), o qual carrega a carga útil da mensagem, tal como valores de sensores do veículo, por exemplo. É importante mencionar que o Campo de Dados não possui um padrão único e a sua codificação pode variar de acordo com o fabricante. Essa codificação específica leva a desafios de interoperabilidade e interpretabilidade universal de mensagens CAN [D'ANDRADA 2020].

O quadro CAN é processado pelas ECUs, que utilizam as informações nele transmitidas para realizar suas funções. Por exemplo, o *cluster* do painel do veículo é controlado por uma ECU que usa os dados das demais ECUs para identificar e sinalizar eventos relevantes.

As redes CAN adotam uma topologia de barramento, comumente conhecida como *CAN Bus*, na qual todos os componentes são conectados por um único fio ou par de fios. Essa topologia permite que todos os componentes se comuniquem entre si e tenham acesso a todas as transmissões realizadas na rede.

No entanto, as redes CAN são amplamente reconhecidas por suas vulnerabilidades significativas, principalmente devido à ausência de mecanismos básicos de segurança, como autenticação e confidencialidade na transmissão de mensagens [Lundberg 2022]. Exemplos de vulnerabilidades incluem ataques de *spoofing*, onde um atacante pode injetar mensagens falsas na rede; ataques de *replay*, onde mensagens capturadas podem ser

retransmitidas para induzir comportamentos indesejados; e ataques de *flooding*, onde um grande número de mensagens é enviado para sobrecarregar o barramento e comprometer a comunicação normal.

Entre as várias possibilidades de ataques cibernéticos, existem modelos em que o atacante atua enviando mensagens com informações falsas para provocar o mau funcionamento das ECUs conectadas à rede. O ataque ***fuzzing***, por exemplo, manipula diversas ECUs com sinais falsos contendo campos de dados e AIDs aleatórios, resultando em falhas no funcionamento do veículo. Esse tipo de ataque é frequentemente escolhido por atacantes que não possuem conhecimento detalhado da rede alvo.

Por outro lado, nos ataques de **fabricação**, as ECUs são manipuladas para agir conforme a intenção do atacante. Isso pode incluir a execução de mensagens com AIDs manipulados ou a modificação do conteúdo dos campos de dados das mensagens. Neste tipo de ataque, o atacante transmite uma mensagem fabricada logo após uma mensagem legítima. A finalidade do ataque pode variar dependendo da abordagem do atacante: utilizando um AID baixo para atrasar uma mensagem real ou injetando uma mensagem maliciosa com o objetivo de causar mau funcionamento nas ECUs.

Em geral, conforme discutido anteriormente, os ataques às redes CAN podem ser realizados de diversas maneiras e direcionados a diferentes ECUs, comprometendo não apenas a segurança das informações, mas também colocando potencialmente em risco a vida dos motoristas. Neste trabalho, serão abordadas as características que podem ser utilizadas por IDSs para identificar corretamente ataques como *fuzzing* e fabricação durante sua execução.

3. Trabalhos Relacionados

IDSs em redes CAN são uma temática complexa e relativamente nova. Além disso, assim como ocorre em outros domínios, existe uma certa falta de confiança nas decisões tomadas por modelos IA empregados na detecção de ataques. Isto evidencia a necessidade de aliar a explicabilidade às decisões tomadas pelo modelo, proporcionando uma visão mais completa do contexto no qual o IDS está inserido. Ainda, o modo como é realizado o treinamento dos modelos e as ferramentas utilizadas para isso — como o *dataset* e o classificador — também podem impactar significativamente a explicabilidade e a confiabilidade do IDS.

Os principais trabalhos correlatos são sumarizados na Tabela 1. Como pode ser observado, há diversos trabalhos que investigam detectores e ataques em redes CAN. Uma parcela significativa dos trabalhos também utiliza XAI para prover as decisões do detector em ataques e conjuntos de dados específicos. Entretanto, é importante destacar que este é o primeiro trabalho a utilizar XAI no conjunto de dados X-CANIDS [Jeong et al. 2024], um *dataset* atual e representativo, analisando ataques de **fabricação** e ***fuzzing***. Adicionalmente, este também é o primeiro trabalho a apresentar interpretações baseadas em engenharia reversa neste contexto.

O trabalho [Swetha et al. 2023] discorre acerca da necessidade de explicabilidade em modelos de IDS baseados em IA. Os autores empregam os algoritmos XGBoost, RF e GB na avaliação de ataques DoS. Ademais, o modelo de predição dos algoritmos empregados é analisado por meio da biblioteca SHAP [ORG 2024]. No entanto, apesar do

Tabela 1. Trabalhos Relacionados

Ref.	CAN	XAI	Interpretação	Detector	Ataques (Dataset)
[Lundberg et al. 2022]	✓	✓	X	DNN	DoS e Fabricação (Survival)
[Le et al. 2023]	✓	✓	X	XGBoost, AdaBoost e GB	Spoofing, Replay, Fuzzing e Flooding (CHADC2020)
[Metwaly and Elhenawy 2023]	✓	✓	X	DNN	Spoofing, DoS e Fuzzing (Car-Hacking)
[Wickramasinghe et al. 2023]	✓	✓	X	RX-ADS (ResNet AE)	DoS e Spoofing (Car-Hacking e OTIDS)
[Hoang et al. 2023]	✓	X	X	CANPerFL	Fuzzing e Replay (Dados Próprios)
[Swetha et al. 2023]	X	✓	X	XGBoost, RF e GB	DoS (NSL-KDD)
[Ding et al. 2024]	✓	✓	X	Deep Learning (multiple)	DoS, Fuzzy e Spoofing (Car-Hacking)
[Jeong et al. 2024]	✓	X	X	X-CANIDS	Fabricação, Masquerade, Suspension e Fuzzing (X-CANIDS)
Este trabalho	✓	✓	✓	**XGBoost**	**Fabricação** e **Fuzzing** (**X-CANIDS**)

emprego de métodos adequados para a XAI, o contexto analisado no trabalho se limita ao escopo das redes tradicionais. Por outro lado, há trabalhos como o de [Hoang et al. 2023], que focam na detecção de intrusões específicas para o domínio particular das redes CAN, mas não empregam nenhuma técnica de XAI. Outra limitação do trabalho consiste no uso de dados privados, os quais não são disponibilizados. Por fim, os autores reconhecem a limitação da ausência de explicabilidade dos resultados de seu trabalho.

A lacuna existente na relação entre explicabilidade e modelos de IDS aplicados às redes CAN é enfatizada em [Le et al. 2023]. Nesse trabalho, o classificador XGBoost é elencado como o que traz as melhores métricas de confiabilidade para o contexto do projeto. Enquanto os autores apresentam um esforço para prover explicabilidade do modelo através da biblioteca SHAP, o conjunto de dados utilizado limita significativamente a efetividade do uso de ferramentas XAI. O trabalho utiliza o conjunto de dados *Car-Hacking Dataset* [Seo et al. 2018], o qual apesar de contemplar amostras de ataques do tipo *DoS*, *Fuzzy* e *Spoofing*, contém apenas 12 atributos. Oito desses atributos são representados por *bytes* cujos significados não são definidos. Portanto, o uso de técnicas de XAI limita-se a mensurar o impacto que esses conjuntos de atributos não definidos causam no modelo, o que pode impactar a confiabilidade [Jeong et al. 2024]. Similarmente, outros estudos, como [Metwaly and Elhenawy 2023], [Ding et al. 2024] e [Wickramasinghe et al. 2023], que utilizam o mesmo conjunto de dados para a avaliação da explicabilidade de IDSs, possuem a mesma limitação de interpretabilidade do real significado das características analisadas.

Já no trabalho de [Wickramasinghe et al. 2023] os autores criam uma abordagem de explicabilidade específica para o modelo proposto. Eles utilizam uma abordagem baseada em aprendizado de máquina adversarial, do inglês *adversarial machine learning*, que exige que novas amostras sejam geradas para explicar o modelo. Essa abordagem torna a proposta limitada, uma vez que para dispor de uma explicabilidade é necessário novas amostras. Além disso, não é possível replicar a explicabilidade proposta em ou-

tros modelos de IDS, devido ao fato dela estar restrita ao contexto do IDS construído no trabalho.

Diferentemente de abordagens anteriores, como [Le et al. 2023], os autores de [Lundberg et al. 2022] direcionam o foco exclusivamente para a explicabilidade das decisões do modelo IDS aplicado às redes CAN, em detrimento de métricas de avaliação de desempenho da IA empregada. O estudo propõe um método de explicabilidade próprio para elucidar as decisões do modelo, que é baseado em DNN (*Deep Neural Network*). Tratar a explicabilidade de maneira a negligenciar métricas de precisão pode ser contraprodutivo, dado que para termos confiança e fidelidade em XAI precisamos de um modelo com métricas razoavelmente precisas. Adicionalmente, o *dataset* survival [Han et al. 2018] utilizado no trabalho compartilha das mesmas limitações anteriormente mencionadas.

Um passo importante rumo à explicabilidade foi dado no trabalho de [Jeong et al. 2024]. Além desse trabalho propor um IDS por meio de um mecanismo próprio, desenvolvido pelos autores, os mesmos coletaram mensagens CAN a partir de um veículo real (Hyundai Sonata 2015) e tornaram essas mensagens publicamente disponíveis através do *dataset* X-CANIDS. Um diferencial significativo desse conjunto de dados para os *datasets Car-Hacking* [Seo et al. 2018] e OTIDS [Lee et al. 2017] consiste no processo de desserialização da carga útil da mensagem CAN, onde dados binários tornam-se sinais legíveis a humanos. Tais dados permitem a criação de um modelo mais robusto, além de contribuir com a explicabilidade. É importante destacar que apesar da maior legibilidade, o significado exato de cada informação não é definido pelos autores do conjunto de dados, os quais argumentam que tais informações não são disponíveis para o público geral, isto é, não é apresentada a *interpretação* dos dados. Ademais, os autores não apresentam a aplicação de técnicas de XAI, como o SHAP [ORG 2024].

Conclui-se, portanto, que a literatura que trabalha IDSs, redes CAN e explicabilidade de maneira simultânea é limitada, pois não contempla a *interpretação*. Desse modo, o presente trabalho propõem-se a abordar tal lacuna por meio da utilização de dados reais e desserializados, conforme providos pelo *dataset* X-CAN-IDS [Jeong et al. 2024].

4. Materiais e Métodos

Uma estratégia amplamente adotada para a implementação de IDSs consiste no uso de algoritmos de aprendizado de máquina, uma subcategoria da IA. No entanto, a utilização desses algoritmos apresenta uma limitação significativa: a falta de interpretabilidade dos modelos gerados. Em outras palavras, interpretar ou explicar por que determinada decisão foi tomada pela IA não é trivial, pois as decisões são baseadas em múltiplos fatores complexos. Como resultado, compreender essas decisões pode se tornar uma tarefa desafiadora.

A IA Explicável surgiu para enfrentar essa limitação, tornando as ações dos modelos mais compreensíveis para os seres humanos [Gunning et al. 2019]. Existem várias ferramentas que auxiliam nesse objetivo, sendo a biblioteca SHAP uma das mais populares [Dwivedi et al. 2023]. O SHAP permite analisar como o modelo chegou a uma decisão, visualizando quais informações foram utilizadas e evidenciando o impacto de diferentes características no processo de tomada de decisão. Dessa forma, é possível gerar diversos gráficos focados em características específicas, contribuindo para a interpretação dos resultados sob diferentes perspectivas [ORG 2024].

Este trabalho foi realizado com base na utilização do *dataset* X-CAN-IDS [Jeong et al. 2024], devido às suas vantagens quando comparado a outros dados publicamente acessíveis, como discutido na Seção 3. Em virtude de sua alta precisão quando aplicado ao contexto em destaque, o algoritmo classificador XGBoost foi o escolhido para o treinamento do modelo. Foram utilizados os hiperparâmetros padrões do algoritmo [Le et al. 2023]. Ademais, de modo a adequar os tipos de dados existentes no *dataset* ao esperado pelo XGBoost, empregou-se biblioteca *LabelEnconder*[1] para transformar os dados *String* em tipos numéricos.

Para o treinamento do nosso modelo, utilizamos um conjunto de dados composto por 392.387 instâncias de dados benignos, concatenado com um conjunto de dados malignos de igual tamanho, que inclui todos os tipos de ataque discutidos anteriormente. Durante o treinamento, aplicamos uma divisão (*split*) de 20% das instâncias (já concatenadas) para a amostra de teste, deixando os 80% restantes para a amostra de treinamento. A quantidade de instâncias foi equilibrada para todos os tipos de ataque. Na Tabela 2, apresentamos de forma mais detalhada a distribuição das instâncias entre as amostras de treinamento e teste.

Tabela 2. Divisão de Instâncias

Total de Instâncias	Amostras de Treinamento	Amostras de Teste
784.744	627.819	156.955

Com o modelo treinado, realizamos a visualização dos resultados utilizando a biblioteca SHAP. Neste trabalho, empregamos o método *explainer*, que utiliza valores SHAP para explicar qualquer modelo de aprendizado de máquina (ML), no nosso caso o XGBoost. Em seguida, geramos diferentes tipos de gráficos fornecidos pela biblioteca SHAP, como os gráficos do tipo *summary plot*, que utilizamos para visualizar a distribuição dos valores SHAP e a importância das características no modelo.

5. Avaliação de Desempenho do XGBoost

Na Figura 2 são apresentados os resultados para o modelo XGBoost treinado para o conjunto de dados X-CANIDS. Como pode ser observado, o modelo atinge resultados muitos bons para todas as métricas, indicando uma elevada eficácia na identificação precisa desses dois tipo de ataques. Entretanto, é importante notar também que o modelo foi ligeiramente melhor para ataques de *fuzzing*.

A sensibilidade é uma métrica que avalia a capacidade do modelo de identificar verdadeiros positivos. Esse valor é crucial para a confiabilidade do IDS, uma vez que uma alta taxa de sensibilidade implica em uma redução na classificação de falsos negativos. Assim, os valores elevados dessa métrica indicam que o modelo é capaz de classificar corretamente a maioria das amostras representando ataques, o que é fundamental para a eficácia de IDSs.

A análise da métrica de precisão, que avalia a proporção de ataques corretamente identificados pelo XGBoost em relação ao total de alertas gerados, é essencial para indicar

[1]Disponível em: `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html`

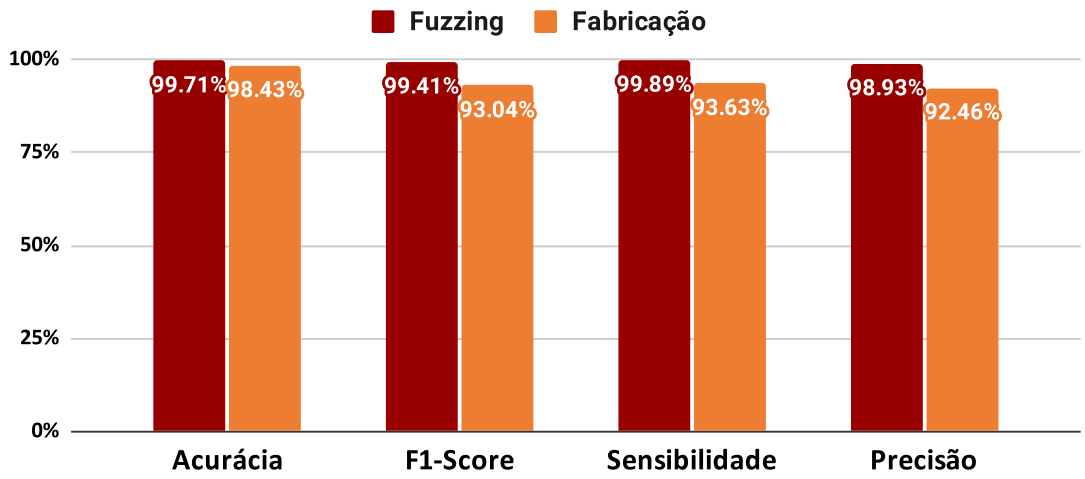

Figura 2. Comparação das métricas avaliadas

baixas taxas de falsos positivos. Os resultados relativamente altos observados para essa métrica impactaram positivamente a explicabilidade do modelo. O modelo atingiu uma alta precisão no conjunto de dados do ataque *fuzzing*, alcançando uma taxa de 98,93%. No caso do ataque de fabricação, embora a precisão tenha sido inferior, o desempenho ainda foi bastante satisfatório, com uma taxa de 92,46%.

Conclui-se, portanto, que o IDS proposto apresenta um bom nível de confiabilidade, tendo em vista os elevados valores obtidos em todas as métricas analisadas. O modelo alcançou uma média de 96,22% para o *F1-Score* e de 99,07% para a acurácia. Esses resultados evidenciam o alto nível de eficácia do modelo na detecção de intrusões no cenário estudado. No entanto, ao se limitar apenas à análise dessas métricas — como frequentemente observado na literatura —, as nuances por trás dos resultados não são plenamente reveladas. Em outras palavras, esses dados carecem de artefatos que sustentem sua interpretabilidade. Portanto, na Seção 6, investigamos a explicabilidade das decisões tomadas pelo modelo. Dessa forma, além de avaliar o desempenho final do IDS, os administradores de rede podem compreender de forma mais detalhada o *modus operandi* dos atacantes.

6. Explicabilidade do Modelo

Com o objetivo de fornecer explicabilidade ao modelo construído com o algoritmo XGBoost, esta seção é dividida em três partes. Primeiramente, na subseção 6.1, utiliza-se a biblioteca SHAP para analisar o impacto das características nas decisões do modelo. Em seguida, na subseção 6.2, os resultados obtidos são cruzados com informações do domínio, permitindo uma explicabilidade mais profunda sobre o modo de operação dos atacantes e os respectivos impactos nos componentes ECUs atacados. Por fim, na subseção 6.3, são apresentados os mapeamentos das ECUs afetadas e os riscos associados.

6.1. Análise dos Valores SHAP

No contexto da análise dos gráficos produzidos através da ferramenta SHAP, é importante destacar que valores vermelhos significam valores altos, enquanto os azuis significam valores baixos. Ademais, no eixo horizontal os valores SHAP são apresentados em uma

escala que começa com valores negativos, passando pelo valor zero e chegando nos valores positivos. Em suma, valores SHAP negativos para uma característica indicam que essa característica contribui para a classificação como *normal* (não intrusão). Valores positivos indicam contribuição para a classificação como *intrusão* (*fuzzing* ou fabricação). Já os valores próximos de zero indicam pouca ou nenhuma influência na decisão do modelo. Os valores SHAP para ambos os ataques estudados são exibidos no gráfico da Figura 3.

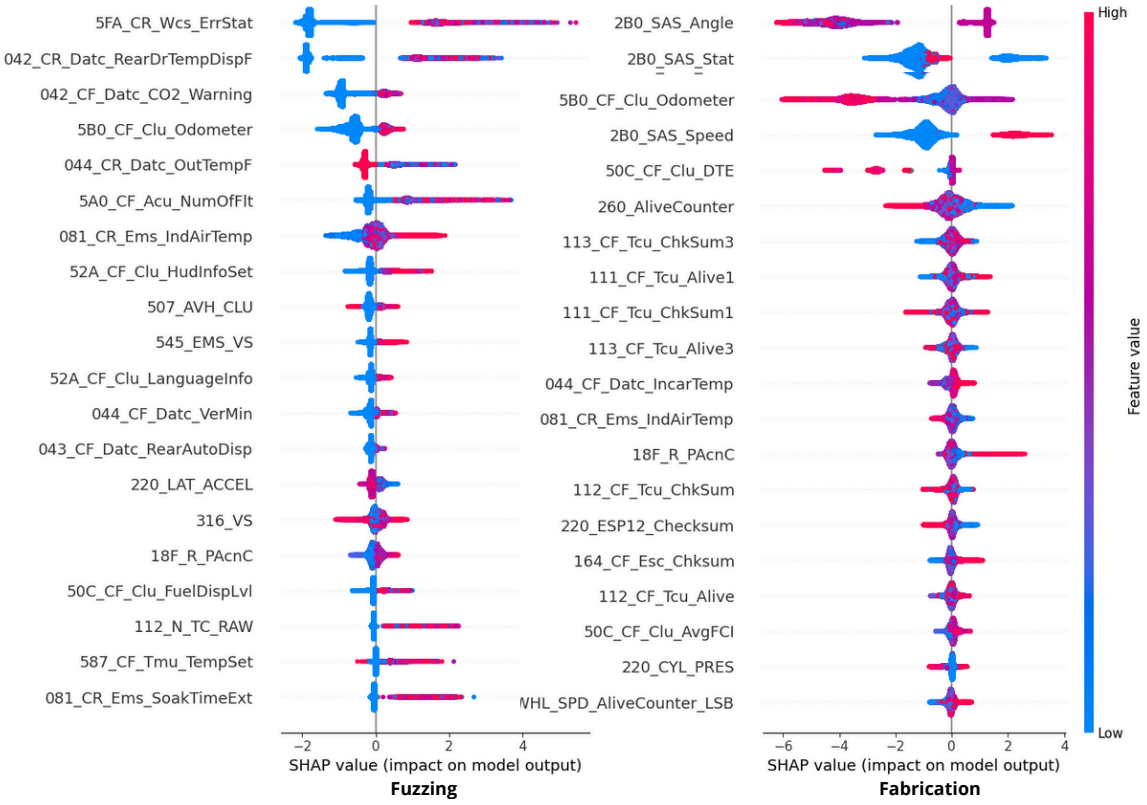

Figura 3. Gráfico *Summary Plot*: Valores SHAP das características.

A análise dos valores SHAP para cada uma das características revela diferenças significativas entre os ataques de *fuzzing* e de fabricação. Vale ressaltar que o conjunto de dados analisado [Jeong et al. 2024] contém 688 características, contudo, o gráfico se limita a exibir apenas aquelas com maior impacto, conforme medido pelos valores SHAP. Das características apresentadas no gráfico, as cinco mais relevantes serão discutidas a seguir.

Com relação às análises do ataque de *fuzzing*, que são apresentadas à esquerda do gráfico, observa-se alguns padrões para as principais características listadas. Primeiramente, as amostras que apresentam valores SHAP negativos possuem valores baixos para as características `5FA_CR_Wcs_ErrStat`, `042_CR_Datc_ReardrTempDispF`, `042_CF_Datc_CO2_Warning` e `5B0_CF_Clu_Odometer`. Isso sugere que tais valores baixos contribuem para a predição da classe normal, evitando especialmente a ocorrência de falsos negativos. Em contraste, a característica `5B0_CF_Clu_Odometer` destaca-se por não apresentar o comportamento inverso, onde seus valores altos estão contribuindo para a predição das amostras de ataques. Ademais, observa-se que para todas essas características há uma zona de sobreposição para os valores SHAP positivos. Isso sugere que

o modelo pode enfrentar dificuldades em diferenciar determinados padrões de ataque, já que as características apresentam tanto valores baixos quanto altos.

Na classe de ataque de fabricação (à direita do gráfico), a `2B0_SAS_Angle` exibe uma predominância de amostras contribuindo para a predição da classe normal, onde seus valores variam de alto a baixo. Observa-se, neste caso, que as amostras de maior influência são os que possuem os valores mais altos para esta característica. Já a característica `2B0_SAS_Stat` exibe um comportamento interessante: seus valores baixos contribuem tanto para a predição da classe normal quanto para a classe de ataque. Em contraste, os valores altos para essa característica, além de contribuírem apenas para a predição de amostras normais, possuem uma contribuição menor do que os valores baixos. Um comportamento diferente é observado pela característica `580_CF_Clu_Odometer`, a qual novamente mostra contribuições mistas, com uma maior tendência para valores SHAP negativos, indicando que seus valores altos estão associados à classe normal.

Uma característica com comportamento que se destaca entre as demais para o ataque de fabricação consiste na característica `2B0_SAS_Speed`. Ela estabelece uma clara distinção entre as amostras com valores baixos contribuindo para a predição da classe normal, enquanto que as amostras com valores altos indicam o contrário. Por fim, a característica `50C_CF_Clu_DTE` apresenta uma distribuição também relevante: as amostras com menor valor SHAP são majoritariamente aquelas com valores altos.

A partir da análise dos valores SHAP, é possível identificar uma clara distinção entre as características impactadas por cada tipo de ataque. Isso sugere que diferentes componentes do veículo podem estar sendo afetados por cada ataque explorado. No entanto, o uso de ferramentas de explicabilidade como o SHAP não garante, por si só, uma verdadeira **interpretabilidade** do significado das informações consideradas relevantes. Para alcançar tal interpretabilidade, é necessário o conhecimento técnico de domínio [Hoang et al. 2023], o que é particularmente desafiador em redes CAN, onde o significado exato das características é confidencial. Ainda assim, este trabalho realiza uma investigação (ver Seção 6.2) para possibilitar a interpretabilidade dos resultados apresentados.

6.2. Interpretabilidade por Engenharia Reversa

Em aplicações de IDSs, o usuário (*e.g.*, administrador de rede) geralmente possui conhecimento sobre as informações sendo analisadas e, portanto, está capacitado para interpretar o significado das características identificadas como relevantes pelas ferramentas de XAI. No entanto, em redes CAN, os IDSs enfrentam uma limitação comum relacionada à interpretabilidade dos dados. Isso ocorre principalmente porque os fabricantes de veículos mantêm em segredo os mapeamentos que permitem a decodificação precisa do conteúdo da carga útil [Hoang et al. 2023]. Além disso, essas definições variam conforme a marca, o modelo e o ano do veículo [Verma et al. 2021]. Implementar um IDS interpretável/explicável, assim como realizar qualquer análise de dados coletados de redes CAN, é de fato um desafio amplamente reconhecido na literatura [Hoang et al. 2023, Verma et al. 2021, Dupont et al. 2019, Shahriar et al. 2023, Jeong et al. 2024].

É importante destacar que o *dataset* X-CANIDS [Jeong et al. 2024] não inclui uma documentação detalhada e completa sobre o conteúdo da carga útil das mensagens. Para enfrentar essa limitação, foi realizado contato via e-mail com os autores do conjunto

de dados, os quais confirmaram a hipótese de que essas informações geralmente são privadas e de difícil acesso ao público. Portanto, a fim de contornar esse desafio e prover interpretabilidade, este trabalho empreendeu um esforço de engenharia reversa. Esse esforço incluiu o mapeamento das ECUs responsáveis pela transmissão de cada sinal, utilizando o documento *CAN Database* [Paul 2021, OpenDBC 2024] e o manual de reparação de componentes do veículo [Hyundai Motor Company 2018a, Hyundai Motor Company 2018b, Hyundai 2024]. Esses documentos foram localizados por meio de investigações conduzidas pelos autores deste trabalho. Assim, a correlação dessas informações permitiu embasar a interpretação das características analisadas. Os resultados dessa análise são discutidos a seguir, considerando-se as características de maior impacto, identificadas a partir dos valores SHAP.

A característica `5FA_CR_Wcs_ErrStat` se destaca de maneira mais evidente e enfática no ataque *fuzzing*. Para o *fuzzing*, valores altos dessa característica (cor vermelha) estão associados a valores SHAP positivos, indicando que essas amostras contribuem para a identificação da classe alvo. Em contraste, valores baixos (cor azul) resultam em valores SHAP negativos, sugerindo uma contribuição para a detecção da classe normal. A investigação realizada neste trabalho revelou que a característica `5FA_CR_Wcs_ErrStat` refere-se ao status de erro do sistema de controle de carga, conhecido em inglês como *Weight Classification System* (WCS) [Hyundai Motor Company 2018b]. Assim, tal informação de fato apresenta um significado em termos de monitoramento de anomalias e, potencialmente, intrusões no sistema. Em se tratando do ataque *fuzzing*, o qual manipula diversas ECUs com sinais falsos a fim de gerar um mau funcionamento do veículo, essa característica traduz o objetivo do atacante.

A característica `5B0_CF_Clu_Odometer` é significativa em ambos os ataques *fuzzing* e fabricação. No ataque *fuzzing*, valores elevados têm um impacto positivo no valor SHAP, auxiliando na detecção da classe alvo. No ataque de fabricação, essa característica também aparece entre as principais, indicando sua importância na identificação de ambos os tipos de ataque. A `5B0_CF_Clu_Odometer` está relacionada ao hodômetro do veículo, fornecendo informações sobre a distância percorrida por ele [Hyundai Motor Company 2018b]. É razoável, portanto, a relação desta informação com ambos os ataques estudados, pois a manipulação dos dados do hodômetro pode ser um indicativo de comportamento anômalo associado a tentativas de ataque. No caso do *fuzzing*, o envio de dados aleatórios ou anômalos ao sistema pode alterar os registros do hodômetro, enquanto na fabricação, a falsificação de informações sobre a quilometragem pode ser usada para mascarar a verdadeira condição do veículo, ambos os casos sendo detectáveis por essa característica.

A característica `042_CR_Datc_RearDrTempDispF` é outra com significância para o ataque *fuzzing*. Os valores altos resultam em valores SHAP positivos, enquanto valores baixos resultam em valores SHAP negativos, indicando sua contribuição para a identificação da classe alvo e da classe normal, respectivamente. Através da sua denominação (`042_CR_Datc_RearDrTempDispF`) é possível inferir que está relacionada à temperatura exibida na porta traseira do sistema de controle climático [Hyundai Motor Company 2018b]. De maneira similar, a característica `044_CR_Datc_OutTempF` é relevante para o ataque *fuzzing*, com valores altos con-

tribuindo positivamente para a predição da classe alvo e valores baixos contribuindo para a classe normal. A `044_CR_Datc_OutTempF`, que é emitida pelo sistema de controle climático, refere-se à temperatura externa e é crucial na utilização do veículo, dado que auxilia no ajuste adequado da temperatura do ambiente interno do automóvel. Tal informação pode ser inferida com base em sua nomenclatura e no manual do veículo [Hyundai Motor Company 2018b]. É, portanto, justificável a relação de ambas as características com o ataque *fuzzing*, pois a manipulação dos dados de temperatura externa pode indicar comportamento anômalo, como a tentativa de confundir o sistema de controle climático e afetar o conforto e a segurança dos ocupantes.

A característica `042_CF_Datc_CO2_Warning` se destaca no ataque *fuzzing*. Valores altos estão associados a valores SHAP positivos, contribuindo para a detecção da classe alvo, enquanto valores baixos resultam em valores SHAP negativos, contribuindo para a detecção da classe normal. Com base em sua nomenclatura e no manual do veículo [Hyundai Motor Company 2018a], é possível inferir que tal característica está associada ao alerta de emissão de CO2 emitido pelo sistema de controle climático. Esse recurso é essencial para a segurança dos ocupantes do veículo. Assim, é plausível assumir que há a relação desta informação com o ataque *fuzzing*, pois a manipulação dos dados de alerta de CO2 pode indicar comportamento anômalo (*e.g.,* tentativas de interferir no funcionamento do sistema climático). Esse ataque pode ser potencialmente perigoso em situações onde o atacante prejudica a detecção de altas concentrações de dióxido de carbono (CO2) dentro da cabine, o que deveria indicar a necessidade de aumentar a ventilação para garantir a segurança dos ocupantes.

Baseado na discussão das características de maior impacto no ataque *fuzzing*, destacam-se principalmente as características relacionadas ao sistema de controle climático [Hyundai Motor Company 2018a], que possuem maior influência e ocorrência de acordo com o nosso gráfico disposto pela ferramenta SHAP através de valores altos. Isso revela que o atacante está operando especialmente nas ECUs relacionadas a esse sistema, sugerindo que as mesmas possam estar comprometidas.

Já para o ataque de fabricação, além da característica `5B0_CF_Clu_Odometer`, a característica `2B0_SAS_Angle` é bastante significativa, com valores altos resultando em valores SHAP positivos. Essa característica refere-se à informação sobre o ângulo do sensor de direção, no inglês *Steering Angle Sensor* (SAS), transmitida através do fluxo identificado por `2B0` [OpenDBC 2024], a qual é fundamental para a estabilidade e controle do veículo. Outra característica importante para a identificação do ataque de Fabricação é a `2B0_SAS_Stat`, onde valores altos estão associados a valores SHAP positivos. A nomenclatura dessa característica sugere que a mesma está relacionada ao status do sensor de direção, assim, sendo crucial para monitorar o desempenho do sistema de direção. Ademais, a característica `2B0_SAS_Speed` também é relevante no ataque de fabricação, com valores altos contribuindo positivamente para a predição da classe alvo. Essa característica está relacionada à velocidade do sensor de direção, importante para o controle dinâmico do veículo [Hyundai 2024].

Adicionalmente, a característica `50C_CF_Clu_DTE` se destaca no ataque de fabricação, com valores altos contribuindo positivamente para a predição da classe alvo. Com base em sua nomenclatura, é possível inferir que essa característica está associada à distância estimada até o esgotamento (DTE) [Hyundai Motor Company 2018b], crucial

para o gerenciamento de energia e eficiência do veículo.

Essa análise mostra a robustez do modelo em utilizar múltiplas características para diferenciar eficazmente entre amostras normais e de ataque, destacando a importância dessas características no contexto dos ataques *fuzzing* e de fabricação. Ademais, como principais conclusões, é possível perceber que os dois modelos de ataques estudados exploram componentes distintos do veículo. Portanto, para além do modo de operação do atacante, é possível estimar que os impactos desses ataques também são diferentes. Por exemplo, foi possível observar que o ataque *fuzzing* está visando, principalmente, as funcionalidades relacionadas ao controle de temperatura e qualidade do ar do veículo. Já a análise do ataque de fabricação demonstra que os principais componentes afetados são aqueles relacionados ao controle de estabilidade do veículo.

6.3. Mapeamento de ECUs e Riscos Associados

Mapear a origem das mensagens maliciosas é fundamental para a contenção do ataque em curso. Com base nas características mais afetadas, foi possível a investigação de quais são as ECUs correspondentes. Na Tabela 3, a relação entre os sinais mais impactantes e suas ECUs de origem é apresentada.

Tabela 3. Relação ECU x Sinal

Ataque	Característica	ECU	Função Principal
Fuzzing	`5FA_CR_Wcs_ErrStat`	ODS	Detecção de Ocupantes
Fuzzing	`042_CR_Datc_RearDrTempDispF`, `044_CR_Datc_OutTempF`, `042_CF_Datc_CO2_Warning`	DATC	Aquecimento, Ventilação e Ar-condicionado
Fuzzing Fabricação	`5B0_CF_Clu_Odometer`	CLU	Painel de Instrumentos
Fabricação	`2B0_SAS_Angle`, `2B0_SAS_Stat`, `2B0_SAS_Speed`	MDPS	Módulo de Direção Assistida
Fabricação	`50C_CF_Clu_DTE`	CLU	Nível de Combustível.

Os ataques de *fuzzing* representam uma ameaça significativa para várias ECUs. A Tabela 3 ilustra a relação entre diferentes ECUs, os sinais que elas transmitem e suas funções principais, destacando as implicações dos ataques a essas ECUs. A análise interpretativa dos valores SHAP resultou em conclusões importantes sobre o comportamento do atacante nos cenários de ataques *fuzzing* e de fabricação.

A ECU de Detecção de Ocupantes (*Occupant Detection System* – ODS) apresenta indícios de que foi comprometida durante o ataque de *fuzzing*, que manipula o sinal `5FA_CR_Wcs_ErrStat`. Este sinal é crucial para a detecção de ocupantes, e ataques podem levar a falsas leituras que afetam a ativação de sistemas de segurança, como *airbags*. A ECU de Aquecimento, Ventilação e Ar-condicionado (*Dual Automatic Temperature Control* – DATC) também está potencialmente em risco, especialmente pelas suspeitas de anomalias nos sinais `042_CR_Datc_RearDrTempDispF`, `044_CR_Datc_OutTempF` e `042_CF_Datc_CO2_Warning`. A manipulação desses sinais pode resultar em desconforto térmico ou falha em alertas de níveis perigosos de dióxido de carbono.

As informações `5B0_CF_Clu_Odometer` e `50C_CF_Clu_DTE`, exibidas no painel de instrumentos, evidenciam o potencial comprometimento da ECU responsável por

esses sinais, a *Cluster Control Unit* (CLU). Isso pode resultar na exibição de dados incorretos sobre a odometria ou o nível de combustível, confundindo o motorista e afetando decisões críticas durante a condução. Ademais, o Módulo de Direção Assistida (*Motor Driven Power Steering* – MDPS) se mostra vulnerável aos sinais `2B0_SAS_Angle`, `2B0_SAS_Stat` e `2B0_SAS_Speed`. Ataques a esses sinais podem influenciar diretamente a direção do veículo, criando situações de condução perigosa.

Entender qual ECU está emitindo mensagens maliciosas é crucial para mitigar esses ataques. Ferramentas XAI, como o SHAP foram vitais nessa análise. O SHAP ajudou a identificar quais características dos dados de entrada mais contribuem para as decisões do modelo de aprendizado de máquina do algoritmo XGBoost, revelando quais ECUs podem estar comprometidas e permitindo a implementação de medidas de segurança apropriadas. Portanto, outro tipo de aplicação interessante de XAI consiste na seleção de características, tal como é explorado em [Scherer et al. 2024]. A aplicação dessas ferramentas é fundamental para aprimorar a segurança automotiva e garantir a integridade dos sistemas críticos dos veículos.

7. Conclusão e Trabalhos Futuros

Este estudo apresentou uma abordagem inovadora para aprimorar a explicabilidade de um IDS, utilizando o algoritmo XGBoost e a biblioteca SHAP no processamento de dados reais coletados de uma rede CAN. Através da análise detalhada dos valores SHAP, foi possível identificar características-chave que contribuem para a classificação de amostras como normais ou intrusivas. Ademais, o mapeamento das ECUs envolvidas, juntamente com a obtenção de informações de referências externas (*e.g.*, manuais do veículo), possibilitou uma melhor interpretabilidade, permitindo a compreensão dos componentes do veículo potencialmente comprometidos por diferentes tipos de ataques.

A investigação conduzida ressalta a importância das ferramentas de XAI na segurança automotiva e na integridade dos sistemas críticos dos veículos. No entanto, a verdadeira explicabilidade dos dados só pode ser alcançada com o conhecimento técnico especializado, algo que este trabalho buscou aprimorar por meio de engenharia reversa e análise de documentação específica.

Ainda assim, permanecem desafios significativos devido à natureza confidencial das informações em redes CAN. Pesquisas futuras podem explorar, por exemplo, a integração de ferramentas de processamento de linguagem natural com XAI, visando facilitar o tratamento de informações e melhorar ainda mais a interpretabilidade dos dados.

Agradecimentos. Esta pesquisa foi parcialmente financiada, com apoio da CAPES – Código de Financiamento 001 e FAPERGS, através dos editais 08/2023 e 09/2023.

Referências

Buscemi, A., Turcanu, I., Castignani, G., Panchenko, A., Engel, T., and Shin, K. G. (2023). A survey on controller area network reverse engineering. *IEEE Communications Surveys & Tutorials*.

D'ANDRADA, L. F. P. (2020). Um sistema de detecção de intrusão de tempo real e baseado em anomalias para redes can automotivas. Master's thesis, Universidade Federal de Pernambuco.

Dhaliwal, S. S., Nahid, A.-A., and Abbas, R. (2018). Effective intrusion detection system using xgboost. *Information*, 9(7).

Ding, W., Alrashdi, I., Hawash, H., and Abdel-Basset, M. (2024). DeepSecDrive: An explainable deep learning framework for real-time detection of cyberattack in in-vehicle networks. *Information Sciences*, 658:120057.

Dupont, G., den Hartog, J., Etalle, S., and Lekidis, A. (2019). A survey of network intrusion detection systems for controller area network. In *2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES)*, page 1–6. IEEE Press.

Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., et al. (2023). Explainable AI (XAI): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9):1–33.

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., and Yang, G.-Z. (2019). XAI-explainable artificial intelligence. *Science Robotics*, 4(37):eaay7120.

Han, M. L., Kwak, B. I., and Kim, H. K. (2018). Anomaly intrusion detection method for vehicular networks based on survival analysis. *Vehicular Communications*, 14:52–63.

Hoang, T.-N., Islam, M. R., Yim, K., and Kim, D. (2023). CANPerFL: Improve in-vehicle intrusion detection performance by sharing knowledge. *Applied Sciences*, 13(11).

Hyundai (2024). Steering angle sensor repair procedures. Accessed: 2024-05-28.

Hyundai Motor Company (2018a). Heater & A/C Control Unit (DATC) repair procedures. Accessed: 2024-05-30.

Hyundai Motor Company (2018b). Hyundai sonata: Trip computer / fuel economy. Acessado em: 28 maio 2024.

Jeong, S., Lee, S., Lee, H., and Kim, H. K. (2024). X-CANIDS: Signal-aware explainable intrusion detection system for controller area network-based in-vehicle network. *IEEE Transactions on Vehicular Technology*, 73(3):3230–3246.

Le, T.-T.-H., Suryanto, N., Kim, H., Ji, J., and Heo, S. (2023). Enhancing intrusion detection and explanations for imbalanced vehicle can network data. In *Proceedings of the 12th International Symposium on Information and Communication Technology*, pages 777–784.

Lee, H., Jeong, S. H., and Kim, H. K. (2017). OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, volume 00, pages 57–5709.

Lokman, S.-F., Othman, A. T., and Abu-Bakar, M.-H. (2019). Intrusion detection system for automotive controller area network (can) bus system: a review. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–17.

Lundberg, H. (2022). Increasing the trustworthiness of AI-based in-vehicle IDS using eXplainable AI.

Lundberg, H., Mowla, N. I., Abedin, S. F., Thar, K., Mahmood, A., Gidlund, M., and Raza, S. (2022). Experimental analysis of trustworthy in-vehicle intrusion detection system using explainable artificial intelligence (XAI). *IEEE Access*, 10:102831–102841.

Metwaly, A. A. and Elhenawy, I. (2023). Sustainable intrusion detection in vehicular controller area networks using machine intelligence paradigm. *Sustainable Machine Intelligence Journal*, 4:(4):1–12.

Moulahi, T., Zidi, S., Alabdulatif, A., and Atiquzzaman, M. (2021). Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus. *IEEE Access*, 9:99595–99605.

OpenDBC (2024). OpenDBC - DBC file basics. Accessed: 2024-05-28.

ORG, S. (2024). Welcome to the SHAP documentation. 16/05/2024.

Paul (2021). DBC 2015 hyundai C-CAN.

Quincozes, S. E., Kazienko, J. F., and Quincozes, V. E. (2023). An extended evaluation on machine learning techniques for denial-of-service detection in wireless sensor networks. *Internet of Things*, 22:100684.

Quincozes, V. E., Quincozes, S. E., Kazienko, J. F., Gama, S., Cheikhrouhou, O., and Koubaa, A. (2024). A survey on IoT application layer protocols, security challenges, and the role of explainable AI in IoT (XAIoT). *International Journal of Information Security*, 23(3):1975–2002.

Scherer, F. H., Dresch, F. N., Quincozes, S. E., Kreutz, D., and Quincozes, V. E. (2024). IWSHAP: Um método de seleção incremental de características para redes CAN baseado em Inteligência Artificial Explicável (XAI). In *Anais do XXIV Simpósio Brasileiro de Segurança da Informaçao e de Sistemas Computacionais*. SBC.

Seo, E., Song, H. M., and Kim, H. K. (2018). GIDS: Gan based intrusion detection system for in-vehicle network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6.

Shahriar, M. H., Xiao, Y., Moriano, P., Lou, W., and Hou, Y. T. (2023). CANShield: Deep-learning-based intrusion detection framework for controller area networks at the signal level. *IEEE Internet of Things Journal*, 10(24):22111–22127.

Swetha, H., R., R. R. R., R., P. R., and Thomas Ciza, B. N. (2023). XAI for intrusion detection system: comparing explanations based on global and local scope. *Journal of Computer Virology and Hacking Techniques*.

Verma, M. E., Bridges, R. A., Sosnowski, J. J., Hollifield, S. C., and Iannacone, M. D. (2021). CAN-D: A modular four-step pipeline for comprehensively decoding controller area network data. *IEEE Transactions on Vehicular Technology*, 70(10):9685–9700.

Wickramasinghe, C. S., Marino, D. L., Mavikumbure, H. S., Cobilean, V., Pennington, T. D., Varghese, B. J., Rieger, C., and Manic, M. (2023). RX-ADS: Interpretable anomaly detection using adversarial ml for electric vehicle CAN data. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):14051–14063.