IWSHAP: Um Método de Seleção Incremental de Características para Redes CAN baseado em Inteligência Artificial Explicável (XAI)

Felipe H. Scherer[1], Felipe N. Dresch[1], Silvio E. Quincozes[1],[2]
Diego Kreutz[1], Vagner E. Quincozes[3]

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

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

[3]Instituto de Computação, Universidade Federal Fluminense (UFF) – Niterói, Brasil

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

{silvioquincozes, diegokreutz}@unipampa.edu.br

vequincozes@midiacom.uff.br

Abstract. *CAN (Controller Area Network) networks are widely used in the automotive industry and are frequently targeted by cyberattacks. Detecting these attacks via machine learning (ML) depends on properly selecting features to ensure the predictive model's performance. This paper proposes IWSHAP, a new feature selection method that combines the Iterative Wrapper Subset Selection (IWSS) algorithm with SHAP values (SHapley Additive exPlanations). Our main goal is to maximize the ML model's performance in a reduced time. The results indicate that IWSHAP can reduce the number of features by up to 99.17% and execution time by 98.3% compared to the baseline.*

Resumo. *As redes CAN (Controller Area Network) são amplamente usadas na indústria automotiva e frequentemente alvo de ataques cibernéticos. A detecção desses ataques via aprendizado de máquina (AM) depende da seleção adequada de características para garantir o desempenho do modelo de predição. Este artigo propõe o IWSHAP, um novo método de seleção de características que combina o algorítimo Iterative Wrapper Subset Selection (IWSS) com os valores SHAP (SHapley Additive exPlanations). O principal objetivo é maximizar a performance do modelo de AM em um tempo reduzido. Os resultados indicam que IWSHAP consegue reduzir o número de características em até 99,17% e o tempo de execução em 98,3% comparado ao baseline.*

1. Introdução

As redes de comunicação CAN (*Controller Area Network*) desempenham um papel crucial em diversos setores, especialmente na indústria automotiva, onde garantem a troca eficiente de dados entre os componentes eletrônicos dos veículos. No entanto, os baixos recursos computacionais utilizados pela rede tornam sua conectividade suscetível a ataques cibernéticos, destacando a necessidade de sistemas robustos de detecção de intrusões que demandem de pouco recurso computacional [Bari et al. 2023, Lokman et al. 2019].

Os Sistemas de Detecção de Intrusões, conhecidos em inglês como *Intrusion Detection Systems* (IDS), que utilizam aprendizado de máquina (AM), emergiram como uma solução promissora, destacando-se na identificação de atividades maliciosas em redes CAN [Lokman et al. 2019, Bari et al. 2023]. No entanto, o desempenho desses sistemas depende fortemente da qualidade dos dados processados, bem como das características selecionadas para o treinamento dos modelos. A seleção apropriada de características representa uma etapa crucial que pode impactar de maneira significativa a eficácia dos sistemas de detecção, além de influenciar diretamente a demanda computacional requerida pelo sistema [Xie et al. 2023, Quincozes et al. 2021].

Diversas técnicas de seleção de características têm sido propostas na literatura, entre elas os métodos de filtro, *wrapping* e *embedded*. Os métodos de filtro, conhecidos pela sua rapidez, baseiam-se na ordenação das características de acordo com sua relevância. No entanto, frequentemente falham em capturar interações complexas entre as características. Em contraste, os métodos *wrapping* oferecem maior precisão, pois levam em consideração essas interações durante o processo de seleção. Todavia, seu elevado custo computacional pode tornar sua aplicação inviável em redes CAN, onde é necessário processar mensagens em tempo real. Já os métodos *embedded* integram a seleção de características diretamente no processo de aprendizagem, sendo específicos para cada algoritmo [Chandrashekar and Sahin 2014, Quincozes et al. 2021].

Recentemente, ferramentas de *eXplainable Artificial Intelligence* (XAI), como a biblioteca SHAP (*SHapley Additive exPlanations*), têm sido amplamente utilizadas para melhorar a compreensão da contribuição de cada característica em modelos de aprendizado de máquina [Došilović et al. 2018, Quincozes et al. 2024]. A SHAP aplica valores baseados na teoria dos jogos para explicar a influência de cada característica no desempenho do modelo, proporcionando novas perspectivas no processo de seleção de características. Especificamente, a SHAP gera um *ranking* das características mais influentes, permitindo uma seleção mais precisa das mais relevantes. Na literatura, diversos estudos têm empregado valores SHAP para a seleção de características, com inspirações provenientes de abordagens de filtragem que utilizam o *ranking* dos valores SHAP [Nazat et al. 2024, E. L. Asry et al. 2024, Setitra et al. 2023]. No entanto, essas abordagens enfrentam desafios semelhantes aos métodos de filtro, como a limitação de capturar interações complexas entre características.

Neste artigo é proposto um novo método de seleção de características, denominado IWSHAP, que combina o algoritmo *Incremental Wrapper Subset Selection* (IWSS) [Bermejo et al. 2009] com valores SHAP. O objetivo é alcançar um equilíbrio entre o desempenho do modelo de AM e o tempo de otimização do conjunto de características analisado. Além disso, é realizada uma avaliação comparativa para determinar se os valores SHAP podem ser eficazes na seleção de características usando uma abordagem de filtro baseada em *ranking*, comparando-a com o método proposto. Para tanto, são executados experimentos a fim de determinar o ponto de corte ideal para o método baseado em *ranking*. Além disso, também é adotado como método comparativo o modelo sem qualquer forma de seleção de características, com o intuito de realizar uma comparação com o IWSHAP e o método de *ranking* a partir dos valores SHAP. Os resultados demonstram que o IWSHAP pode contribuir significativamente para a detecção de intrusões em redes CAN, oferecendo uma solução mais balanceada e eficiente para a seleção de ca-

racterísticas. O método se destaca tanto em capacidade de redução (*i.e.*, menor número de características selecionadas) quanto em termos de tempo de execução. Ademais, o IWSHAP gera também as melhores métricas de desempenho do classificador.

Este trabalho está organizado como segue. A Seção 2 cobre a fundamentação teórica, enquanto a Seção 3 discute os trabalhos relacionados. O algoritmo do IWSHAP é detalhado na Seção 4. As Seções 5 e 6 descrevem os experimentos realizados e os resultados obtidos, respectivamente. Finalmente, a Seção 7 traz as conclusões do estudo.

2. Fundamentação Teórica

As redes veiculares desempenham um papel fundamental na comunicação interna de componentes automotivos, sendo cruciais para a segurança e eficiência dos veículos modernos. Uma das tecnologias-chave nestas redes é o protocolo CAN, que permite a comunicação direta entre os componentes do veículo sem a necessidade de um *host* central [Jeong et al. 2024]. A comunicação nesta rede é realizada por meio do quadro CAN, que é o padrão de mensagens da rede CAN. Ainda, o protocolo CAN define quatro tipos diferentes de quadros com diferentes usos, mas apenas o quadro de dados é utilizado para transmitir dados entre as Unidades de Controle Eletrônico (*Electronic Control Unit* – ECU). O quadro de dados é composto por diversos atributos. Dentre esses atributos, se destaca o identificador arbitrário (AID) que é fixo e informa a prioridade da mensagem, o código de comprimento dos dados (DLC) e o *data payload* (dados) onde as mensagens de fato estão contidas [Pollicino et al. 2024].

Este protocolo é valorizado por sua capacidade de reduzir a complexidade do sistema automotivo e por oferecer uma confiabilidade robusta, essencial para operações críticas. Contudo, a segurança dessas redes é de suma importância, pois falhas podem comprometer não apenas a funcionalidade do veículo, mas também a segurança física dos usuários. Assim, a implementação de medidas de segurança robustas no design de redes veiculares é importante para prevenir potenciais vulnerabilidades [Lokman et al. 2019].

Para proteger essas redes, os Sistemas de Detecções de Intrusões, do inglês, *Intrusion Detection Systems* (IDS) são implementados [Jeong et al. 2024, Lokman et al. 2019]. Eles monitoram o tráfego e as atividades do sistema em tempo real, utilizando algoritmos para detectar padrões anormais que possam indicar uma intrusão. Recentemente, IDSs têm incorporado técnicas de aprendizado de máquina para aprimorar a detecção de ameaças. Essa detecção baseia-se na análise de dados como assinaturas de *malware*, padrões de tráfego de rede e outras heurísticas para diferenciar atividades legítimas de maliciosas [Quincozes et al. 2024].

No entanto, a eficácia dos IDSs é amplamente determinada pela qualidade das características dos dados que eles analisam. A seleção de características desempenha um papel crucial na otimização da performance desses sistemas. Essa técnica visa identificar as características mais relevantes para a detecção eficaz de intrusões, descartando aquelas que são redundantes ou menos importantes. Isso reduz a carga computacional e aumenta a precisão do modelo. Existem diversas abordagens para essa seleção, incluindo: métodos de filtro, que se baseiam em estatísticas para escolher as características; métodos de *embedding*, que integram a seleção ao processo de aprendizado de máquina; e métodos de *wrapping*, que utilizam ciclos iterativos para testar e selecionar os melhores subconjuntos de características [Quincozes et al. 2021].

Dentro dos métodos de *wrapping*, o *Incremental Wrapper Subset Selection* (IWSS) é um algoritmo que destaca-se por sua eficiência e adaptabilidade. O IWSS realiza uma seleção incremental de subconjuntos, mantendo apenas as características que apresentam uma melhoria na função de aptidão do modelo preditivo [Quincozes et al. 2021]. Esse processo iterativo continua até que a adição de novas características não resulte em melhorias significativas no desempenho do modelo ou até que um número predefinido de características seja alcançado. Além disso, o IWSS considera as interações entre características, o que contribui para a construção de modelos mais precisos e robustos.

Ademais, nos últimos anos, a *eXplainable Artificial Intelligence* (XAI) tem ganhado destaque como um campo essencial para aumentar a confiança e a compreensão humana nos sistemas de IA. Dentro desse contexto, o método SHAP (*SHapley Additive exPlanations*) se destaca por sua capacidade de fornecer explicações claras e detalhadas sobre o impacto de cada característica nas decisões dos modelos de IA. Utilizando princípios da teoria dos jogos, o SHAP atribui um valor numérico a cada característica, representando sua contribuição para a decisão final do modelo. Isso permite identificar quais características são mais influentes, e, além disso, como elas interagem para influenciar o resultado do modelo, tornando as decisões mais compreensíveis [Quincozes et al. 2024]. Em redes CAN, técnicas como o SHAP permitem a interpretabilidade do modo de operação dos atacantes a fim de mapear quais ECUs são afetadas [Dresch et al. 2024].

A aplicação do SHAP em conjunto com técnicas de seleção de características, como o algoritmo IWSS, tem o potencial de transformar a eficácia dos IDSs em redes veiculares CAN. Esta combinação não só refina a seleção de características, assegurando que apenas as mais relevantes sejam consideradas, mas também fornece uma análise detalhada de como essas características influenciam na detecção de ameaças.

3. Trabalhos Relacionados

A seleção de características utilizando recursos de XAI é um tema relativamente recente e que tem chamado a atenção em diferentes contextos [Khani et al. 2024, Yang et al. 2023]. Entretanto, diversos trabalhos realizam a seleção de características baseada em *rankings* apresentados por ferramentas de XAI, utilizando os resultados na eliminação das características menos relevantes considerando limiares de corte arbitrários definidos pelos autores [Setitra et al. 2023, Nazat et al. 2024, Roshan and Zafar 2021]. No entanto, tratar o processo de seleção dessa forma pode levar a vieses e ineficiência dos modelos, uma vez que as interações entre as características podem acabar sendo descartadas.

Como pode ser observado na Tabela 1, existem trabalhos recentes que utilizam XAI na seleção de características para IDSs em diferentes contextos, como CAN, IoT e redes de computadores de maneira geral. Entretanto, pode-se constatar que a maioria dos trabalhos focam apenas na abordagem de filtro para a seleção de características e poucos trabalhos utilizam XAI nesse processo de seleção para o domínio das redes CAN.

Alguns dos trabalhos não utilizam XAI no processo de seleção de características. Por exemplo, o [Aksu and Aydin 2022] reconhece a necessidade de realização do processo de seleção de características da forma adequada, propondo um novo método de seleção do tipo *wrapper* denominado MGA (*Modified Genetic Algorithm*). Os autores utilizam deste método para identificar o melhor subconjunto de dados, trazendo maior

Tabela 1. Trabalhos Relacionados

Trabalho	Seleção com XAI	Método de seleção	Domínio	Tipo de Ataque
[Aksu and Aydin 2022]	X	Wrapping	CAN	Vários (Injeção, DoS, Exploit, etc.)
[Ullah et al. 2022]	X	Filtragem	CAN, IoV	Spoofing, Fuzzing, DDoS
[Mowla et al. 2022]	X	Filtragem	CAN	Masquerade, Malfunction, Fuzzy, Flooding, Fabricação
[Nazat et al. 2024]	SHAP	Filtragem	Genérico	Dos, Sybil, Falsificação
[Setitra et al. 2023]	SHAP	Filtragem	SDIoT	DDoS
[Roshan and Zafar 2021]	SHAP	Filtragem	Genérico	DDoS
[Bhandari et al. 2020]	SHAP	Filtragem	IoT	Impersonation, Injeção, Flooding
[E. L. Asry et al. 2024]	SHAP	Filtragem	Genérico	DoS, Probe, R2L, U2R
Este Trabalho	**SHAP**	**Filtragem & *Wrapping***	**CAN**	**Suspensão**

otimização para o IDS. Ainda, os autores expõem em suas conclusões a melhora significativa nas métricas utilizadas sob o IDS após a seleção de características. Apesar disso, esse trabalho não utiliza métodos de XAI e um dos *datasets* utilizados, o Car-Hacking [Seo et al. 2018], deixa a desejar na sua cobertura de ataques [Lee et al. 2023]. A evidência de melhoras no IDS após o processo de seleção de características também é exposta no trabalho [Ullah et al. 2022], no qual os autores realizam, após um pré-processamento dos dados, a filtragem das características com a finalidade de encontrar aquelas mais relevantes. Já em [Mowla et al. 2022] a interpretabilidade é atingida através da exploração de características e modelos. O processo de exploração de características é baseado em estatística univariada, que permite a identificação das características estatisticamente mais relevantes. O estudo destaca o aumento da confiabilidade e do desempenho do modelo, o que deve estar entre os objetivos de métodos de seleção de características.

Autores de trabalhos como [E. L. Asry et al. 2024], [Roshan and Zafar 2021], [Bhandari et al. 2020] e [Setitra et al. 2023] adotam abordagens similares. Por exemplo, em [Setitra et al. 2023] os autores propõe um IDS aplicado ao contexto de IoT (*Internet of Things*) e especializado em ataques de DDoS (*Distributed Denial of Service*). A seleção de características é realizada através da abordagem de filtro baseada na importância encontrada através dos valores SHAP. Entretanto, os autores não especificam o limiar de filtragem utilizado no processo e nem a quantia de características reduzidas.

Já o trabalho [Roshan and Zafar 2021] segue a mesma abordagem e informa os limiares e a quantidade de características selecionadas. No trabalho foram utilizados apenas dados benignos na fase de treinamento e na fase de teste foi empregada uma combinação de dados benignos com dados de ataque *DDoS*. Os autores treinam três modelos distintos: o primeiro com todas as 78 características do *dataset*, o segundo apenas com as características com uma taxa de correlação menor que 0,8 e terceiro utilizando as melho-

res 30 características baseadas nos valores SHAP. Os resultados indicam que utilização dos valores SHAP na seleção de características leva a modelos com melhores métricas de classificação e também a uma redução no consumo de recursos do IDS.

No trabalho [E. L. Asry et al. 2024], a seleção de características também foi realizada com base nos valores SHAP para os dois *datasets* utilizados (NSL-KDD [Tavallaee et al. 2009] e UNSW-NB15 [Moustafa and Slay 2015]), que englobam as seguintes categorias de ataques: *DoS* (*Denial of Service*), Probe, R2L (*Remote-to-Local*) e U2R (*User-to-Root*). O modelo de classificação utilizado foi o XGBoost e os resultados encontrados se alinham com os demais trabalhos que empregaram a mesma técnica.

Os autores de [Nazat et al. 2024] propõem um IDS para sistemas de direção autônomas, o que se assemelha ao contexto de redes CAN. O principal objetivo dos autores foi testar diferentes modelos de aprendizado de máquina e identificar o mais adequado ao contexto. A etapa de seleção de características foi parte crucial desse processo, mas baseada essencialmente em três limiares diferentes de valores SHAP para os *datasets* utilizados no trabalho. Apesar de efetiva em alguns contextos, a simplicidade dos métodos baseados em limiares entra em conflito com a complexa natureza de modelos de aprendizado de máquina. Os próprios autores afirmam que a seleção de características baseada exclusivamente em limiares pode prejudicar alguns modelos, sendo necessário considerar outros aspectos na eliminação de características [Fryer et al. 2021].

Em [Roshan and Zafar 2022] os autores combinam a utilização dos valores SHAP e do modelo SVM para selecionar os quarenta principais recursos do conjunto de dados. Apesar de levar a seleção de características a resultados de classificação promissores, o principal problema desta abordagem está relacionado ao elevado custo computacional e complexidade intrínseca do SVM. Buscando uma abordagem que gere bons resultados a um baixo custo computacional, neste trabalho utilizamos como base o IWSS, um algoritmo que se destaca por sua eficiência e adaptabilidade.

4. Algoritmo Proposto

O pseudo-código do IWSHAP é apresentado no Algoritmo 1. O método de seleção de características é voltado para redes CAN e utiliza os valores SHAP combinados com o algoritmo *Iterative Wrapper Subset Selection* (IWSS). Essa combinação potencializa, ao mesmo tempo, uma seleção de qualidade e um baixo consumo de recursos computacionais, algo fundamental em uma rede CAN que opera com processamento de mensagens em tempo real. Uma descrição mais detalhada da ferramenta construída a partir do método proposto neste trabalho está disponível em [Scherer et al. 2024].

O processo do método inicia com a divisão dos conjuntos de treinamento e teste, utilizando a proporção de 80% dos dados para treinamento e 20% para teste e a definição do parâmetro `random_state` em 42 para garantir a reprodutibilidade dos resultados (Linha 1). A seguir, é instanciado um modelo XGBoost (Linhas 2 e 3).

Na linha 4 do algoritmo é instanciada a técnica XAI denominada SHAP para calcular a importância das características, criando um *ranking* das mais influentes na tomada de decisão para o modelo. Esta etapa é seguida da criação de um novo *dataframe* seguindo o *ranking* criado dos valores SHAP mais relevantes (Linhas 5 a 7).

O próximo estágio do método consiste em uma iteração pelo conjunto de carac-

Algorithm 1 Pseudoalgoritmo do Modelo de Classificação

1: $(X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}}) \leftarrow$ train_test_split$(X, y, 0.2, 42)$
2: model $\leftarrow$ XGBoost(use_label_encoder $= 0$, eval_metric $=' logloss'$)
3: model.fit$(X_{\text{train}}, y_{\text{train}})$
4: shap_values $\leftarrow$ shap.TreeExplainer(model).shap_values(X_{train})
5: importancia_df $\leftarrow$ np.abs(shap_values).mean(0)
6: importancia_df $\leftarrow$ pd.DataFrame({'feature': X_{train}.columns, 'importance': importancia})
7: importancia_df $\leftarrow$ importancia_df.sort_values('importance', asc $= 0$)
8: best_features $\leftarrow \{\}$, best_f1_score $\leftarrow 0$
9: **for** $i \leftarrow 1$ to len(importancia_df) **do**
10: current_features $\leftarrow$ best_features $\cup$ {importancia_df['feature'][i]}
11: $X_{\text{train\_selected}}, X_{\text{test\_selected}} \leftarrow X_{\text{train}}$[current_features], X_{test}[current_features]
12: model_selected $\leftarrow$ XGBoost(use_label_encoder $= 0$, eval_metric $=' logloss'$)
13: model_selected.fit$(X_{\text{train_selected}}, y_{\text{train}})$
14: y_pred_selected $\leftarrow$ model_selected.predict$(X_{\text{test_selected}})$
15: f1 $\leftarrow$ f1_score$(y_{\text{test}}$, y_pred_selected$)$
16: **if** f1 $>$ best_f1_score **or** f1 $== 0$ **then**
17: (best_f1_score, best_features) $\leftarrow$ (f1, current_features)
18: **end if**
19: **end for**
20: (best_features_final, best_f1_score_final) $\leftarrow$ (best_features, best_f1_score)

terísticas mais influentes (Linhas 8 a 19). O processo inicia com um conjunto vazio de melhores características (Linha 8) e segue até iterar e comparar todas as características. O primeiro passo da iteração consiste em testar o conjunto. Para isso, realiza-se o treinamento do modelo e calcula-se sua métrica de F1-Score. Esta métrica será utilizada para comparar a qualidade do conjunto atual com a do conjunto anterior. Caso seja a primeira iteração, o conjunto atual será considerado o melhor, tornando-se a referência para as iterações subsequentes (Linhas 10 a 15).

Na etapa de comparação é averiguado se o conjunto atual de características possui um *F1-Score* maior que o anterior (Linhas 16 a 18). Caso essa condição seja aceita, o melhor conjunto e o o melhor *F1-Score* são atualizados. Além disso, na etapa de comparação, o F1-Score igual a zero (Linha 16) é utilizado como critério para identificar o melhor conjunto. Essa abordagem deve-se ao fato de que uma característica isolada não permite o cálculo de métricas, pois possui pouca relevância para o modelo quando considerada individualmente —– mesmo aquelas que estão no topo do *ranking*. Por conseguinte, essa condição permanece válida até que se encontre o primeiro conjunto de características com uma métrica F1-Score calculável.

Em resumo, um novo conjunto é mantido se a sua pontuação *F1-Score* for melhor do que a do conjunto anterior. Caso isso aconteça, o conjunto e sua métrica são atualizados e o processo continua até que todas as características dispostas no *dataframe* tenham sido avaliadas. Por fim, o conjunto final das melhores características e as melhores métricas são obtidos, proporcionando uma seleção de características eficiente e balanceada para a detecção de intrusões.

5. Configuração Experimental

Os experimentos foram realizados em um servidor com processador AMD Ryzen 7 5800X 8-Core, 64 GB de memória RAM e rodando o sistema operacional Ubuntu 22.04. Para a realização dos experimentos, utilizamos diversas tecnologias, métodos e abordagens utilizados como parâmetro comparativo no qual será discorrido nesta seção.

Especificamente, empregamos o explicador SHAP projetado para modelos de árvores, que tem a capacidade de realizar explicações em um período reduzido de tempo quando comparado ao explicador padrão SHAP [ORG 2024]. Ainda, a escolha do algoritmo XGBoost para a construção do modelo de detecção de intrusão se deu por sua comprovada eficiência e alta performance em tarefas de classificação [Dhaliwal et al. 2018].

Para avaliar o IWSHAP foi utilizado conjunto de dados X-CANIDS, coletado de um Hyundai LF Sonata 2017 e-VGT em ambientes reais [Jeong et al. 2024]. A seleção do X-CANIDS foi motivada principalmente pelo fato de conter dados reais que, após sua extração, passaram pelo processo de desserialização, tornando-os legíveis para humanos. De acordo com esses autores, esse procedimento de desserialização aumenta significativamente as métricas do modelo [Jeong et al. 2024]. Os autores também argumentam que esse *dataset* pode ser considerado um dos mais completos e atualizados disponíveis para redes CAN.

O *dataset* contém mensagens CAN e está disponível nos formatos de mensagens brutas e sinais desserializados, contendo um total de 688 características por amostra. O conjunto de dados inclui os ataques *Fuzzing*, fabricação, suspensão, *Masquerade* e repetição. Entretanto, neste experimento foram utilizados apenas os ataques de suspensão realizados através do AID `2B0h` em razão de sua métrica *F1-Score* extremamente baixa (0,07%) nos experimentos realizados pelo autor. Ainda, foi utilizado um total de 784.744 instâncias de dados, sendo 392.387 instâncias de dados benignos e 392.387 de dados malignos provenientes do ataque de suspensão concatenados em um *dataframe*, onde posteriormente utilizou-se da biblioteca `LabelEnconder` para converter os valores categóricos em numéricos.

Para avaliar o desempenho dos modelos, foram utilizadas as métricas de *F1-Score*, *Recall*, *Precision*, tempo de execução e quantidade de características selecionadas. Ademais, foram definidos três modelos adicionais para avaliar comparativamente o desempenho do IWSHAP. O primeiro consiste em um modelo sem nenhum tipo de seleção de características, ao qual classificamos como *baseline*. O segundo modelo leva em conta a seleção de características com base no *ranking* de valores SHAP. Finalmente, o algoritmo IWSS foi empregado como base para o terceiro modelo.

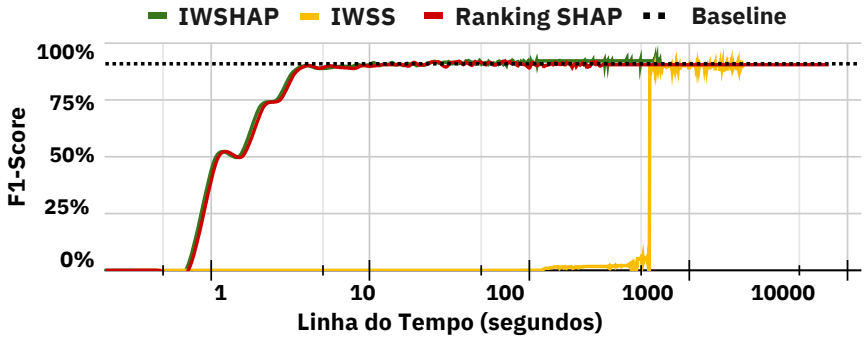
6. Resultados e Discussões

Nesta seção, apresentamos os resultados dos experimentos realizados para avaliar a eficácia do método IWSHAP em comparação com outras abordagens de seleção de características, como IWSS, *ranking* SHAP e *baseline*. Primeiramente, discutimos o desempenho do IWSHAP em termos de detecção de intrusões (Seção 6.1), focando em métricas como *F1-Score*, *Recall* e *Precision*. Em seguida, analisamos o uso de recursos computacionais, como CPU e memória, para cada método (Seção 6.2). Por fim, exploramos a capacidade do IWSHAP de identificar e interpretar ataques em redes CAN (Seção 6.3),

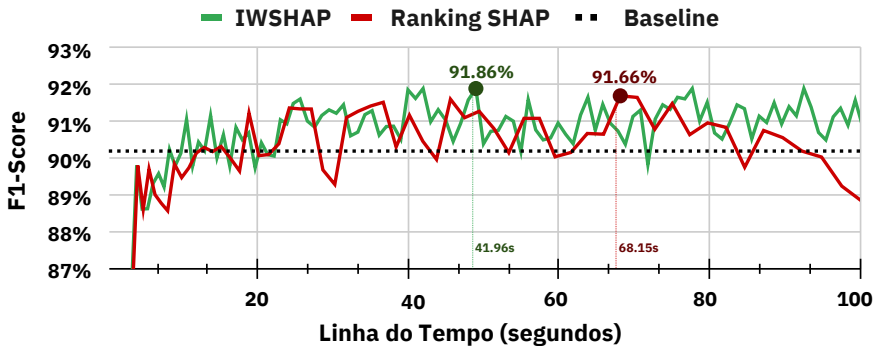
destacando a contribuição das características selecionadas para a melhoria da segurança do sistema. Os resultados demonstram que o IWSHAP não só melhora o desempenho de detecção, mas também mantém um uso eficiente de recursos computacionais, tornando-o uma solução robusta e eficaz para a seleção de características em redes CAN.

6.1. Análise de Desempenho na Detecção de Intrusões

Para realizar uma avaliação comparativa da proposta apresentada, o IWSHAP foi comparado com as abordagens *IWSS*, *ranking SHAP* e *baseline*. A eficácia do IWSHAP foi comprovada através de métricas como o tempo de convergência na análise dos conjuntos de características, o tempo computacional do conjunto reduzido resultante, e as métricas de detecção (*F1-Score*, *Recall* e *Precision*).



(a) Comparação de tempo de execução total x F1-Score (Zoom 1)



(b) Comparação de tempo de execução total x F1-Score (Zoom 2)

Figura 1. Comparação de tempo de execução por rodada x F1-Score em diferentes níveis de zoom

A Figura 1(a) expõe a linha do tempo das três diferentes abordagens, representando o tempo total do experimento até chegar ao melhor resultado em termos de *F1-Score*. A partir dessa análise, é possível constatar a ligeira vantagem do IWSHAP quando comparado com as demais abordagens. Já a Figura 1(b) permite visualizar de maneira ampliada a vantagem do IWSHAP, que levou 26.19 segundos a menos para atingir um *F1-Score* 0,20% maior em comparação com a abordagem *ranking* SHAP. Ressalta-se, ainda, a ausência da abordagem do IWSS deste gráfico, tendo em vista que a linha do tempo dessa abordagem se encontra fora do recorte proposto, que facilita a visualização. O IWSS só atingiu seu melhor *F1-Score* após mais de 1000 segundos de execução e, ainda assim, sendo inferior ao IWSHAP.

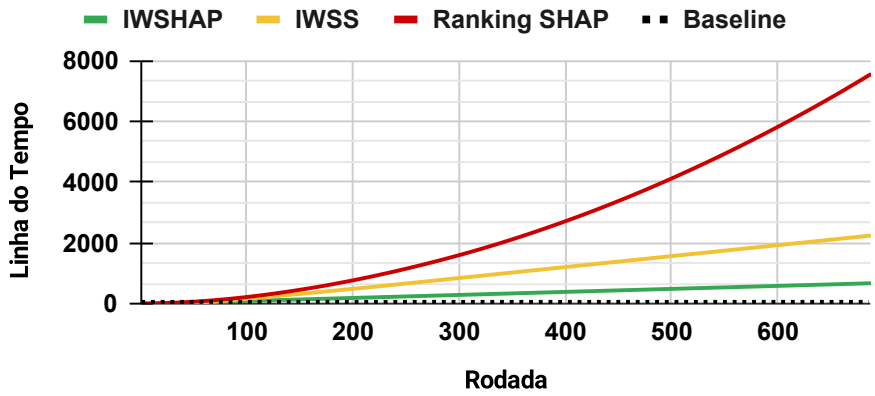


Figura 2. Relação do tempo total decorrido (Linha e o número da rodada.

Complementarmente, a Figura 2 permite observar a linha do tempo de cada abordagem, destacando-se o IWSHAP pela sua notável eficiência na redução de tempo total de execução. Ao observar a linha do tempo (eixo Y), é possível percebermos que na rodada 600 o IWSHAP fica mais de 60% abaixo do IWSS, por exemplo.

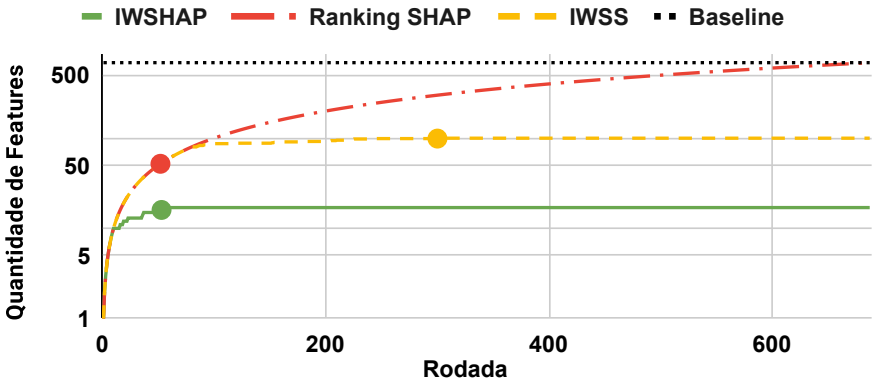


Figura 3. Ponto de convergência (maior F1-Score).

Na Figura 3, a relação de quantia de características utilizadas em cada rodada é demonstrada, onde as marcações ilustradas pelos círculos evidenciam o momento em que cada abordagem alcançou sua melhor métrica de *F1-Score*. Enquanto que o *baseline* analisa 688 características a implementação do IWSS estabiliza em apenas 100 características. Com a utilização da seleção por *ranking* SHAP, esse número é reduzido para 52 características. Entretanto, é importante ressaltar que o conjunto de características foi ainda mais drasticamente reduzido com a utilização do método IWSHAP, indo para apenas 16. Essa redução representa uma diminuição de 69,23% na quantia de características em relação ao melhor resultado obtido pela abordagem de seleção por *ranking* SHAP e uma redução de 99,17% quando comparado a *baseline*, que utilizou de 688 características. Ademais, é importante destacar ainda que, com relação ao ponto de convergência (Figura 2), em termos de tempo de execução, o IWSHAP obteve o melhor tempo, sendo 55,5%, 71,89% e 98,3% mais eficiente que os algoritmos SHAP, IWSS e a *baseline*, respectivamente.

O algoritmo IWSHAP também obteve desempenho consistentemente superior em

outras métricas de desempenho, conforme evidenciado pela Figura 4. Evidencia-se, portanto, a superioridade do IWSHAP em termos de *F1-Score*, *Recall* e *Precision*. Dessa forma, conclui-se que o algoritmo proposto é comprovadamente superior no que tange o tempo de execução, quantia de características e métricas de desempenho do classificador.

Figura 4. Comparação das métricas avaliadas para o melhor conjunto (ponto de convergência).

6.2. Análise de Uso de Recursos Computacionais

Ao analisar o uso computacional, apresentado nas Figuras 5 e 6, observa-se que o IWSHAP consome ligeiramente mais CPU que o IWSS, com uma diferença pouco significativa de menos de 3%. É natural e esperado que o IWSHAP utilize mais CPU que o IWSS, pois combina também uma avaliação de valores SHAP, o que acrescenta uma carga computacional adicional. No entanto, ambos os métodos apresentam um uso de CPU consideravelmente superior ao *baseline*, indicando que os métodos de seleção de características introduzem uma sobrecarga computacional significativa.

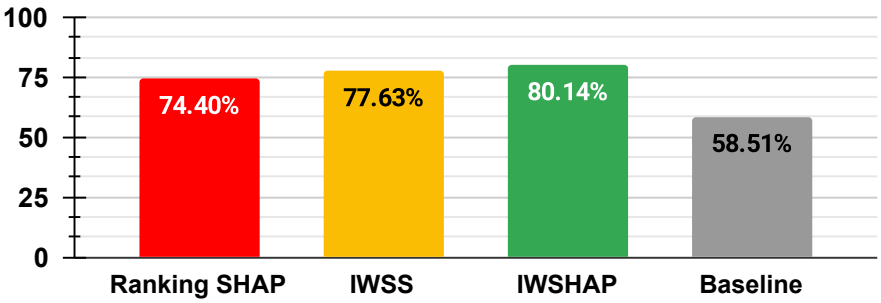


Figura 5. Uso máximo de CPU

Além disso, a Figura 6 mostra que o consumo de memória do IWSHAP é muito similar ao do IWSS, com uma diferença de apenas 0,8% a favor do IWSHAP. Esse resultado sugere que, apesar da adição da avaliação de valores SHAP, o IWSHAP consegue manter um consumo de memória eficiente, equiparando-se ao IWSS e mantendo-se próximo ao *baseline*. Isso é relevante, pois o uso de memória é um fator crítico em muitas aplicações práticas, especialmente aquelas que operam em ambientes com recursos limitados.

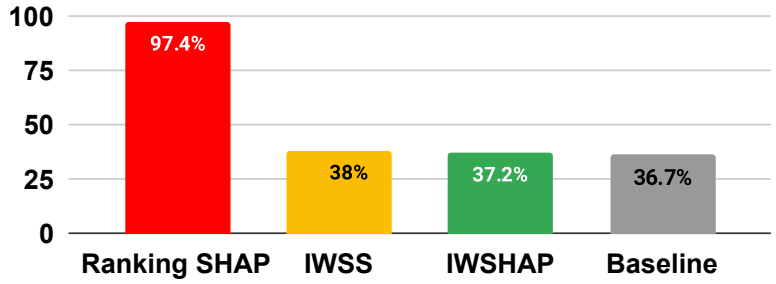


Figura 6. Uso máximo de memoria.

Dessa forma, os resultados indicam que, embora o IWSHAP introduza um aumento no uso de CPU, ele consegue manter o uso de memória em níveis competitivos com o IWSS. Já em termos de tempo de execução, o IWSHAP destaca-se por seu tempo de execução significativamente menor, levando apenas 1,01 segundos para atingir o ponto de convergência. Em comparação, o IWSS requer 3,59 segundos e o *ranking* SHAP 2,22 segundos, mostrando que ambos são consideravelmente mais lentos que o IWSHAP.

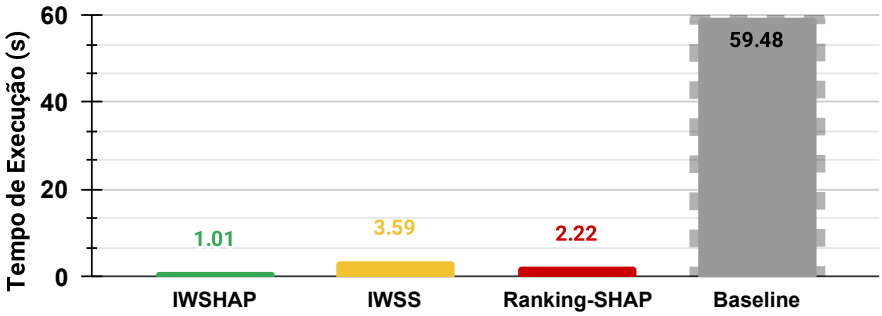


Figura 7. Tempo de execução para o melhor conjunto (ponto de convergência).

O *baseline*, que não utiliza nenhum método de seleção de características, apresenta um tempo de execução extremamente elevado de 59,48 segundos. Esses resultados evidenciam a eficiência do IWSHAP, não apenas em termos de precisão na seleção de características, mas também na economia de tempo, o que é crucial para aplicações em tempo real e ambientes com recursos computacionais limitados.

6.3. Interpretação de Ataques usando IWSHAP

A partir das características selecionadas pelo IWSHAP, foi possível identificar de quais ECUs os sinais comprometidos estavam vindo. Na Figura 8, as ECUs coloridas representam aquelas que foram comprometidas.

As ECUs comprometidas identificadas são: *Sistema de Freios Antibloqueio (ABS)*, responsável por evitar o bloqueio das rodas durante frenagens; *Sistema de Gerenciamento do Motor (EMS)*, que controla os parâmetros de funcionamento do motor; *Direção Assistida por Motor Elétrico (MDPS)*, que facilita a direção do veículo; *Unidade de Painel de Instrumentos (CLU)*, que exibe informações críticas para o motorista; *Controle Eletrônico de Estabilidade (ESC)*, que ajuda a manter a estabilidade do veículo; e *Unidade de Controle da Transmissão (TCU)*, que gerencia o sistema de transmissão do veículo. Iden-

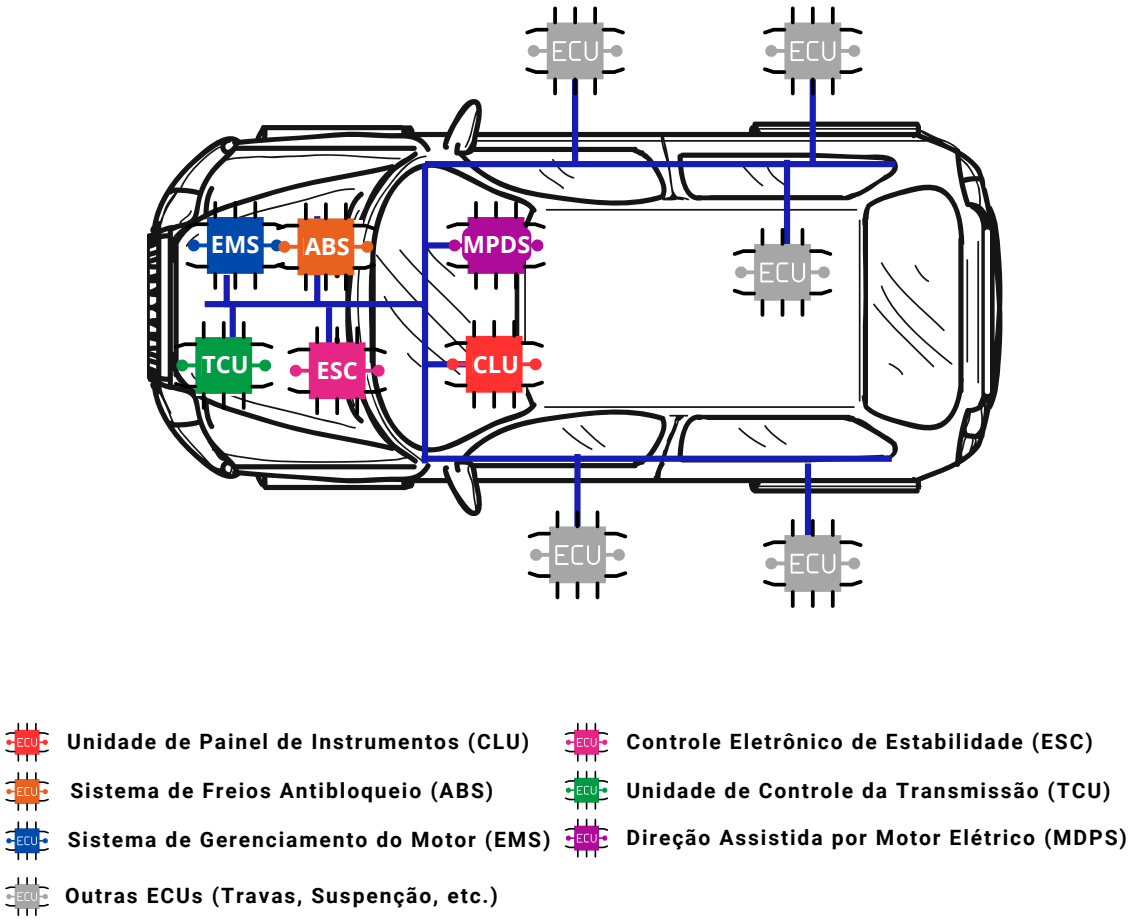


Figura 8. ECUs comprometidas.

tificar essas unidades comprometidas é crucial para uma resposta eficaz a incidentes de segurança e para garantir a integridade e a segurança do sistema automotivo.

Embora o propósito original do SHAP seja justamente prover tal interpretabilidade através da sua abordagem de XAI, o emprego do IWSHAP permite uma visão mais precisa ao empregar o IWSS nas características ordenadas pelo valor SHAP. Isso se deve ao uso do método *Wrapping*, onde a performance do classificador XGBoost adotado revela de fato quais são as informações que o atacante manipula. Com isso, ao rastrear a origem dessas informações, foi possível identificar as ECUs comprometidas.

O método IWSHAP combina a capacidade de interpretabilidade do SHAP com uma seleção de características mais refinada proporcionada pelo IWSS, oferecendo um *insight* mais claro sobre a origem dos ataques. No contexto de redes CAN, onde múltiplas ECUs controlam diferentes aspectos do veículo, a habilidade de identificar precisamente quais unidades estão sendo alvo de um ataque é crucial para a segurança e a resposta eficaz a incidentes.

Portanto, a combinação de SHAP com IWSS não só identifica quais características

são mais importantes, mas também de quais ECUs essas características se originam, permitindo uma compreensão mais profunda dos ataques. Desse modo, o IWSHAP permite a identificação precisa das ECUs comprometidas, como ilustrado na figura.

No ambiente automotivo, onde a segurança e a integridade dos sistemas são vitais, a capacidade de detectar e interpretar ataques com precisão é essencial. O IWSHAP proporciona essa capacidade, tornando-se uma ferramenta valiosa para engenheiros e especialistas em segurança de veículos.

7. Considerações Finais

Este estudo apresentou o método IWSHAP, uma abordagem inovadora para a seleção de características em redes CAN utilizando XAI. O método combina o algoritmo IWSS com valores SHAP para otimizar a seleção de características, maximizando o desempenho do modelo de aprendizado.

Os principais resultados obtidos indicam que o IWSHAP não só melhora o *F1-Score*, *Recall* e *Precision* dos modelos de detecção de intrusões, mas também reduz significativamente o tempo de execução e a quantidade de características necessárias para a análise. A redução de 99,17% no número de características comparado ao *baseline* e uma diminuição de 98,3% no tempo de execução demonstram a eficiência e a eficácia do método proposto. Essas melhorias são cruciais para a aplicação prática em redes CAN, onde o processamento em tempo real e a limitação de recursos computacionais são desafios constantes.

Como trabalhos futuros, podem ser exploradas a aplicação do IWSHAP em outros contextos além das redes CAN, como em sistemas IoT e redes de comunicação críticas. Ademais, futuras pesquisas podem almejar a integração do método com outras técnicas de XAI e explorar a combinação deste com diferentes algoritmos de aprendizado de máquina para avaliar seu desempenho em uma variedade de cenários de detecção de intrusões. A implementação do IWSHAP em ambientes reais e sua validação com diferentes *datasets* também são passos importantes para consolidar a aplicabilidade e a robustez do método proposto. Finalmente, estudos futuros podem considerar também aspectos técnicos de otimização profunda do método para reduzir ainda mais o consumo de CPU e memória.

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

Aksu, D. and Aydin, M. A. (2022). MGA-IDS: Optimal feature subset selection for anomaly detection framework on in-vehicle networks-CAN bus based on genetic algorithm and intrusion detection approach. *Computers & Security*, 118:102717.

Bari, B. S., Yelamarthi, K., and Ghafoor, S. (2023). Intrusion detection in vehicle controller area network (CAN) bus using machine learning: A comparative performance study. *Sensors*, 23(7).

Bermejo, P., Gámez, J. A., and Puerta, J. M. (2009). Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection. In *2009 IEEE symposium on computational intelligence and data mining*, pages 367–374. IEEE.

Bhandari, S., Kukreja, A. K., Lazar, A., Sim, A., and Wu, K. (2020). Feature selection improves tree-based classification for wireless intrusion detection. In *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, SNTA '20, page 19–26, New York, NY, USA. Association for Computing Machinery.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & electrical engineering*, 40(1):16–28.

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

Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE.

Dresch, F. N., Scherer, F. H., Quincozes, S. E., and Kreutz, D. L. (2024). Modelos interpretáveis com inteligência artificial explicável (XAI) na detecção de intrusões em redes intra-veiculares controller area network (CAN). In *Anais do XIX Simpósio Brasileiro de Segurança da Informaçao e de Sistemas Computacionais*. SBC.

E. L. Asry, C., Benchaji, I., Douzi, S., and E. L. Ouahidi, B. (2024). A robust intrusion detection system based on a shallow learning model and feature extraction techniques. *PLOS ONE*, 19(1):1–31.

Fryer, D., Strümke, I., and Nguyen, H. (2021). Shapley values for feature selection: The good, the bad, and the axioms. *Ieee Access*, 9:144352–144360.

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.

Khani, P., Moeinaddini, E., Abnavi, N. D., and Shahraki, A. (2024). Explainable artificial intelligence for feature selection in network traffic classification: A comparative study. *Transactions on Emerging Telecommunications Technologies*, 35(4):e4970.

Lee, S., Choi, W., Kim, I., Lee, G., and Lee, D. H. (2023). A comprehensive analysis of datasets for automotive intrusion detection systems. *Computers, Materials & Continua*, 76(3):3413–3442.

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.

Moustafa, N. and Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6.

Mowla, N. I., Rosell, J., and Vahidi, A. (2022). Dynamic Voting based Explainable Intrusion Detection System for In-vehicle Network. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 406–411.

Nazat, S., Li, L., and Abdallah, M. (2024). XAI-ADS: An explainable artificial intelligence framework for enhancing anomaly detection in autonomous driving systems. *IEEE Access*, 12:48583–48607.

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

Pollicino, F., Stabili, D., and Marchetti, M. (2024). Performance comparison of timing-based anomaly detectors for controller area network: A reproducible study. *ACM Trans. Cyber-Phys. Syst.*, 8(2).

Quincozes, S. E., Mossé, D., Passos, D., Albuquerque, C., Ochi, L. S., and dos Santos, V. F. (2021). On the performance of GRASP-based feature selection for CPS intrusion detection. *IEEE Transactions on Network and Service Management*, 19(1):614–626.

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.

Roshan, K. and Zafar, A. (2021). Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation (SHAP). *International Journal of Computer Networks Communications (IJCNC)*, 13(6):109–128.

Roshan, K. and Zafar, A. (2022). Using kernel SHAP XAI method to optimize the network anomaly detection model. In *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 74–80.

Scherer, F. H., Dresch, F. N., Quincozes, S. E., Kreutz, D., and Quincozes, V. E. (2024). IWSHAP: Uma ferramenta para seleção incremental de características utilizando IWSS e SHAP. In *Anais Estendidos 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.

Setitra, M. A., Fan, M., and Bensalem, Z. E. A. (2023). An efficient approach to detect distributed denial of service attacks for software defined internet of things combining autoencoder and extreme gradient boosting with feature selection and hyperparameter tuning optimization. *Transactions on Emerging Telecommunications Technologies*, 34(9):e4827.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6.

Ullah, S., Khan, M. A., Ahmad, J., Jamal, S. S., e Huma, Z., Hassan, M. T., Pitropakis, N., Arshad, and Buchanan, W. J. (2022). HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles. *Sensors*, 22(4).

Xie, J., Sage, M., and Zhao, Y. F. (2023). Feature selection and feature learning in machine learning applications for gas turbines: A review. *Engineering Applications of Artificial Intelligence*, 117:105591.

Yang, Z., Wang, Z., Huang, C., and Yao, X. (2023). An explainable feature selection approach for fair machine learning. In *International Conference on Artificial Neural Networks*, pages 75–86. Springer.