Uma Avaliação Estendida do Impacto da Seleção e Enriquecimento de *Features* em Sistemas de Detecção de Intrusão para *Smart Grids*

Vagner E. Quincozes[1], Silvio E. Quincozes[2], Célio Albuquerque[1], Diego Passos[3], Daniel Mossé[4]

[1] Universidade Federal Fluminense - UFF

[2]Universidade Federal do Pampa - UNIPAMPA

[3]Instituto Superior de Engenharia de Lisboa - ISEL

[4]Universidade de Pittsburgh - PITT

vequincozes@midiacom.uff.br, silvioquincozes@unipampa.edu.br,

celio@ic.uff.br, diego.passos@isel.pt, mosse@pitt.edu

Abstract. *This work evaluates the impact of feature selection and enrichment on the performance of intrusion detection systems (IDS) for smart grids. Seven feature sets were tested, ranging from basic to enriched versions, including two application orders: (i) selection after enrichment and (ii) enrichment after selection. Effectiveness was assessed across seven types of cyberattacks with varying complexity, using lightweight classifiers. The results show that feature selection improves detection in simpler attacks, such as Random Replay and Inverse Replay, while enrichment enhances performance in more complex scenarios, such as Masquerade Fake Fault. The best results were obtained by combining both techniques, especially when enrichment was applied before selection — which helped preserve critical derived features, such as delay.*

Resumo. *Este trabalho avalia o impacto da seleção e do enriquecimento de features no desempenho de IDSs para smart grids. Foram testados sete conjuntos de features, de versões básicas a enriquecidas, incluindo duas ordens de aplicação: (i) seleção após enriquecimento e (ii) enriquecimento após seleção. A eficácia foi analisada em sete tipos de ciberataques com diferentes complexidades, por meio de classificadores leves. Os resultados mostram que a seleção melhora ataques simples, como Random Replay e Inverse Replay, enquanto o enriquecimento se destaca em cenários mais complexos, como Masquerade Fake Fault. Os melhores ganhos ocorreram com a combinação das duas técnicas, especialmente quando o enriquecimento foi aplicado antes da seleção — o que evitou a perda de features derivadas importantes, como delay.*

1. Introdução

Os sistemas ciberfísicos, do inglês *Cyber-Physical Systems* (CPS), desempenham um papel crucial em setores críticos como energia, transporte e manufatura, integrando componentes físicos e digitais para otimizar operações em larga escala. No contexto de substações elétricas digitais, ou *smart grids*, por exemplo, esses sistemas possibilitam uma gestão eficiente e remota da geração, distribuição e consumo de energia. Contudo, essa interconexão aumenta a exposição a ciberataques, que podem comprometer tanto a

integridade das operações quanto a segurança física de infraestruturas críticas. Um exemplo foi o ataque coordenado pelo grupo SandWorm, afiliado à inteligência militar russa, que comprometeu sistemas de 22 empresas de infraestrutura crítica na Dinamarca, incluindo o setor de energia, forçando interrupções nas operações e destacando os riscos associados à crescente interconectividade desses sistemas [Technologies 2024].

Nesse cenário, os sistemas de detecção de intrusão (do inglês *Intrusion Detection Systems* - IDS) são fundamentais para proteger infraestruturas críticas, monitorando redes, identificando anomalias e mitigando ameaças [Thakkar and Lohiya 2022, Horchulhack et al. 2022]. No entanto, para lidar com a complexidade dos ciberataques, que cresce exponencialmente, fez-se necessário o uso de aprendizado de máquina, que é bastante custoso. Duas abordagens promissoras visando aumentar a eficiência dos IDSs na classificação do perfil das atividades analisadas [Quincozes et al. 2024a] são a seleção de *features*, que identifica as *features* mais relevantes e elimina redundâncias nos dados, e o enriquecimento de *features*, que adiciona novas *features* a partir daquelas existentes.

Embora a seleção e o enriquecimento de *features* sejam amplamente reconhecidos como técnicas promissoras para aumentar a eficácia dos IDSs [Ngo et al. 2024], ainda existem incertezas sobre como aplicá-las de forma sistemática. Existem estudos que avaliam os benefícios da seleção de *features* [Zouhri et al. 2024, Rahim and Manoharan 2024, Kaushik et al. 2023], estudos que avaliam o enriquecimento de *features* [Sarhan et al. 2024, Musleh et al. 2023, Quincozes et al. 2024c] e ainda aqueles que comparam os benefícios da seleção *versus* enriquecimento [Li et al. 2024, Ngo et al. 2024]. No entanto, ainda é necessário avaliar como essas abordagens se comportam em conjunto, particularmente se uma delas é suficiente para alcançar resultados satisfatórios ou se sua combinação é necessária em cenários mais complexos. Questões como a ordem de aplicação — seleção antes do enriquecimento ou o inverso — e os impactos em diferentes cenários permanecem sem respostas claras.

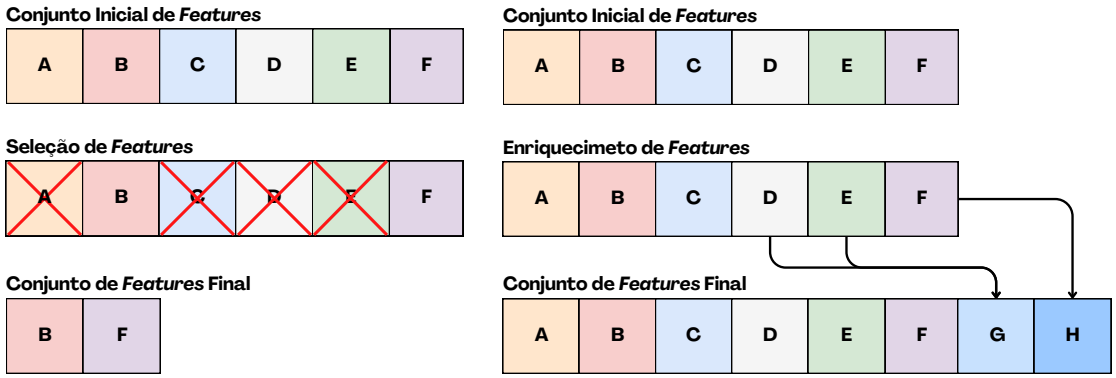
Para responder essas questões, esse trabalho propõe uma análise aprofundada do impacto da seleção e do enriquecimento de *features* no desempenho de IDSs em sistemas ciberfísicos, com ênfase em *smart grids*. As principais contribuições deste estudo são:

1. Uma avaliação extensiva envolvendo sete conjuntos de *features*, inclusive aqueles obtidos a partir dos processos de enriquecimento e seleção de *features*;
2. Investigação dos processos de seleção e enriquecimento de *features* em múltiplos tipos de ataques a fim de analisar o impacto desses processos;
3. A demonstração de como a interação entre seleção e enriquecimento de *features* pode otimizar o desempenho de IDS em ataques complexos, oferecendo *insights* para a construção de IDS mais eficientes; e
4. Integração de uma análise de explicabilidade com a ferramenta SHAP (*SHapley Additive exPlanations*) para identificar causas de desempenho inesperado, revelando limitações da seleção univariada de *features* e orientando futuras melhorias.

O restante deste trabalho está organizado da seguinte forma: A Seção 2 apresenta a fundamentação teórica. A Seção 3 discute os trabalhos relacionados. Na Seção 4, são detalhados os aspectos metodológicos adotados para a análise proposta. A Seção 5 apresenta os resultados obtidos e as respectivas discussões. A Seção 6 analisa resultados fora dos padrões por meio de uma ferramenta de XAI. Por fim, a Seção 7 apresenta a conclusão e direções para pesquisas futuras.

2. Fundamentação Teórica

Em CPSs, como *smart grids*, a integração de dispositivos físicos e cibernéticos amplia significativamente a superfície de ataques, tornando os IDSs uma necessidade crítica [Mariani et al. 2024]. Nesses ambientes, além de monitorar padrões de tráfego e atividades de sistemas, IDSs precisam identificar anomalias que possam indicar tentativas de intrusão. A complexidade dos protocolos específicos para *smart grids*, tais como GOOSE (*Generic Object Oriented Substation Event*) e SV (*Sampled Values*), definidos pelo padrão IEC-61850 [Mackiewicz 2006], exige técnicas mais avançadas (*e.g.*, aprendizado de máquina) para que os IDSs consigam operar eficientemente. Em particular, a qualidade das *features* utilizadas desempenha um papel crucial, uma vez que *features* representativas são essenciais para a eficácia de um IDS [Ngo et al. 2024], mas *features* demais podem confundir os algoritmos. Duas abordagens de pré-processamento que são amplamente utilizadas para otimizar as *features* processadas por IDSs são a seleção de *features* e o enriquecimento de *features*, conforme ilustrado na Figura 1.



(a) Seleção de *Features*. (b) Enriquecimento de *Features*.

Figura 1. Processos de Seleção e Enriquecimento de *Features*.

A seleção de *features* (Figura 1(a)) permite identificar as *features* mais relevantes e eliminar redundâncias ou dados com ruídos que podem prejudicar o desempenho dos modelos. Por exemplo, o GRASPQ-FS [Quincozes et al. 2024b] é uma metaheurística eficaz para seleção de *features*, combinando exploração e refinamento na busca por soluções otimizadas. Sua abordagem inclui duas fases principais: uma fase de construção, que gera soluções iniciais, e uma fase de busca local, que refina essas soluções para melhorar seu desempenho. Uma particularidade do GRASPQ-FS consiste no uso de uma fila de prioridade, que organiza as soluções (conjuntos de *features*) com base em critérios de qualidade. Esse mecanismo reduz significativamente o custo computacional ao limitar o número de iterações durante a busca por novas soluções, mantendo apenas aquelas mais promissoras para fase de busca local, que é computacionalmente custosa devido ao grande número de soluções vizinhas a serem avaliadas. Especificamente, o número de iterações é reduzido para *iterações na fase de construção* × N, onde N é o tamanho da fila de prioridade. Isso torna o GRASPQ-FS especialmente adequado para cenários de alta dimensionalidade.

O enriquecimento de *features* (Figura 1(b)), por outro lado, visa ampliar a utilidade dos dados por meio da geração de novas *features* derivadas de informações básicas [Xi et al. 2024]. O objetivo é fornecer ao sistema informações adicionais que

capturem padrões mais complexos, permitindo maior precisão na detecção de intrusões, especialmente em cenários sofisticados e dinâmicos. Esse processo pode ser realizado por meio de diferentes abordagens, cada uma projetada para destacar aspectos específicos dos dados. Duas técnicas frequentemente utilizadas em sistemas ciberfísicos são a análise interprotocolo e a correlação temporal [Quincozes et al. 2024a]. A análise interprotocolo integra informações de múltiplos protocolos, combinando *features* que, quando analisados separadamente, não revelariam interações significativas. Por exemplo, combinar *timestamps* de diferentes protocolos pode expor atrasos ou inconsistências causadas por ataques que exploram falhas de sincronização. Já a correlação temporal consiste em identificar padrões e relações ao longo do tempo em uma mesma *feature* ou em múltiplas *features*, permitindo que o IDS reconheça anomalias que se desenvolvem em função de mudanças temporais, como variações de frequência ou atrasos em sinais consecutivos.

Outra abordagem de manipulação de *features* é a extração de *features*, que transforma o espaço de *features* com o objetivo de reduzir a dimensionalidade do conjunto de dados, mantendo ou sintetizando a maior parte da informação relevante [Sarhan et al. 2024]. Diferente da seleção, que mantém apenas as *features* mais importantes do conjunto original, e do enriquecimento, que gera novas *features* a partir das existentes, a extração constrói novas *features* com base em combinações ou transformações dos dados originais, frequentemente descartando as *features* iniciais [Latif et al. 2025]. Apesar de suas diferenças, a extração pode ser complementar ao enriquecimento: enquanto o enriquecimento gera *features* mais expressivas para capturar comportamentos complexos, a extração pode ser usada para condensar informações redundantes. No entanto, ela não está no escopo deste artigo.

3. Trabalhos Relacionados

A literatura sobre IDSs explora técnicas de manipulação de *features*, agrupadas em três categorias principais: (i) seleção, (ii) enriquecimento ou extração, e (iii) comparações entre essas abordagens.

Diversos estudos destacam a importância da seleção de *features* para melhorar a eficiência dos IDSs. Por exemplo, [Rahim and Manoharan 2024] propuseram o modelo Spiking VGG-16, que utiliza *Skill Optimization Algorithm* (SOA) para selecionar *features* relevantes, demonstrando o impacto positivo dessa abordagem. De forma semelhante, o modelo ZESO-DRKFC [Rabie et al. 2022] também utiliza uma técnica de seleção baseada em otimização (*Zaire Ebola Search Optimization* - ZESO) para identificar *features* importantes, aumentando a precisão e reduzindo falsos positivos em sistemas SCADA. Outros estudos, como o de [Kaushik et al. 2023], avaliaram diferentes técnicas de seleção, como *Chi-Square* e *Information Gain*, comparando seus efeitos em classificadores tradicionais, enquanto [Zouhri et al. 2024] realizaram uma análise do impacto de filtros univariados e multivariados, como *Double Input Symmetric Relevance* (DISR) e *Correlation-based Feature Subset Selection* (CFS), demonstrando que filtros multivariados são mais eficazes para manter ou melhorar o desempenho com conjuntos reduzidos de *features*. Esses trabalhos enfatizam que a seleção de *features* é uma técnica eficiente para otimizar IDSs, mas não abordam como combiná-la com técnicas de enriquecimento ou extração, o que representa uma lacuna na literatura.

Além desses trabalhos, estudos recentes investigaram técnicas de enrique-

cimento e extração de *features* como estratégias para melhorar o desempenho de IDSs. [Sarhan et al. 2024] exploraram algoritmos de extração como Principal Component Analysis (PCA), *Linear Discriminant Analysis* (LDA) e *Auto-Encoder*, destacando que o desempenho dessas técnicas varia de acordo com o *dataset* utilizado, sem uma abordagem universalmente superior. De forma complementar, [Musleh et al. 2023] analisaram um dataset de tráfego de rede convertido em imagens binárias e utilizaram redes neurais profundas, como DenseNet e VGG-16, para extrair *features* automaticamente. Essa abordagem permitiu capturar padrões complexos a partir das representações visuais dos pacotes de rede, alcançando uma acurácia de até 98,3%. Já [Quincozes et al. 2024c] focaram no enriquecimento de *features* para redes IEC-61850, criando 39 novas características interprotocolo e temporais, o que resultou em um aumento no F1-score de 95,6% para 99,4%. Recentemente, [Xi et al. 2024] introduziram o IDS-MTran, que utiliza um módulo de enriquecimento cruzado de *features* (*Cross Feature Enrichment*) para integrar informações em múltiplas escalas, causando melhorias no desempenho ao capturar interações complexas entre níveis de abstração. Esses estudos ressaltam a eficácia de técnicas avançadas de extração e enriquecimento, mas geralmente as aplicam de forma isolada, sem explorar como sua integração com seleção de *features* pode potencializar os resultados.

Por fim, existem alguns trabalhos que realizam análises comparativas entre as técnicas de seleção e extração de *features* para IDSs [Li et al. 2024, Ngo et al. 2024]. Especificamente, [Li et al. 2024] avaliaram essas técnicas no *dataset* TON-IoT, demonstrando que a extração apresenta melhor desempenho em cenários com poucas *features*, enquanto a seleção é mais eficiente computacionalmente em conjuntos maiores de dados. Já em [Ngo et al. 2024], foi realizada uma análise no *dataset* UNSW-NB15, destacando que a seleção de *features* oferece maior acurácia e menor tempo de processamento para valores altos de K (*i.e.*, número de *features* reduzidas), enquanto a extração se mostrou mais robusta na detecção de ataques variados em cenários com valores baixos de K. Embora ambos os trabalhos forneçam diretrizes práticas para a escolha entre essas abordagens, eles não exploram as potenciais vantagens de combinar seleção e extração de *features*.

4. Metodologia

Neste trabalho, o *dataset* IEC-61850 [Quincozes et al. 2024a] foi adotado, o qual inclui amostras de tráfego normal e ataques às *Smart Grids* (Tabela 1). A divisão de amostras por classe e a proporção de treinamento e teste foram preservados, mantendo as características do *dataset* original para representar situações reais de tráfego de rede. Esse dataset foi escolhido por ser recente, conter um grande volume de amostras, diversos tipos de ataque e *features* enriquecidas bem documentadas.

O *dataset* contém um total de 2.955.738 amostras no conjunto de treinamento e 2.955.648 no conjunto de teste, sendo cada amostra caracterizada por 69 *features*. As *features* são divididas em 35 básicas e 34 enriquecidas. As *features* básicas incluem 22 extraídas diretamente do protocolos GOOSE e 13 provenientes do protocolo SV. As 34 *features* enriquecidas resultam de correlações temporais e interprotocolo, com 10 *features* enriquecidas a partir do GOOSE e 24 do SV. A *feature* `stNum` do protocolo GOOSE, por exemplo, origina a *feature* enriquecida `stDiff`, que calcula a diferença entre valores consecutivos da `stNum`. Além disso, algumas *features* enriquecidas combinam informações de múltiplas *features* individuais ou protocolos. Por exemplo, a *feature* `delay` é calculada como a diferença entre o `gTimestamp` do GOOSE e o `time` do

Classe/Ataque	Treinamento	Teste
Normal (sem ataque)	2.759.425	2.755.139
Random Replay	39.000	39.000
Injection	39.000	39.000
High StNum	39.000	39.000
Inverse Replay	26.033	30.319
Poisoned High Rate	18.574	18.570
Masquerade Fake Normal	17.419	17.420
Masquerade Fake Fault	17.287	17.200

Tabela 1. Classes e Número de Amostras do *Dataset* IEC-61850.

SV, representando um comportamento interprotocolo. A relação completa das *features* e seus respectivos processos de enriquecimentos estão na Figura 2 e disponíveis no trabalho original que propõe o *dataset* IEC-61850 [Quincozes et al. 2024a].

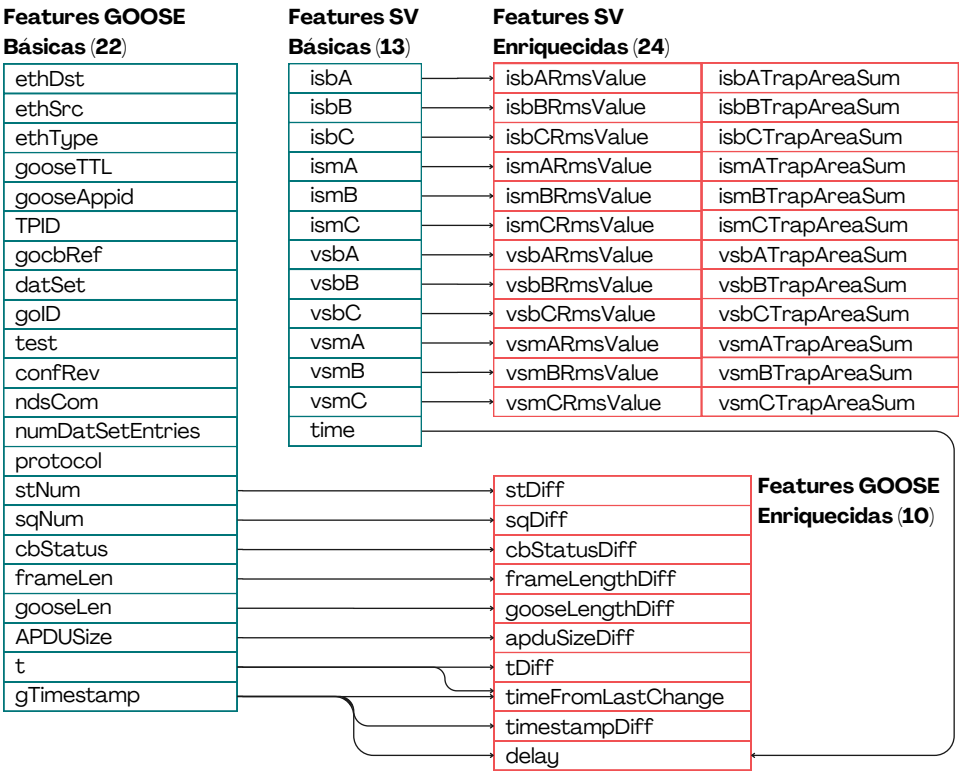


Figura 2. *Features* Básicas e Enriquecidas do *Dataset* IEC-61850.

O fluxo de experimentos adotado neste trabalho compreende três etapas principais, conforme a Figura 3: Pré-processamento, *Pipeline* de análise e Avaliação. No pré-processamento, os dados do *dataset* IEC-61850 são preparados, incluindo a normalização das colunas numéricas, a codificação das colunas categóricas e a configuração de análises binárias para cada classe de ataque (*i.e.,* cada ataque foi avaliado isoladamente, mantendo-se as mesmas amostras normais para todos os experimentos). Para cada *dataset* reduzido resultante, foram gerados três conjuntos de *features*, incluindo GOOSE básicas (22 features), GOOSE + SV básicas combinadas (35 features) e GOOSE + SV básicas enriquecidas (69 features) para fins comparativos.

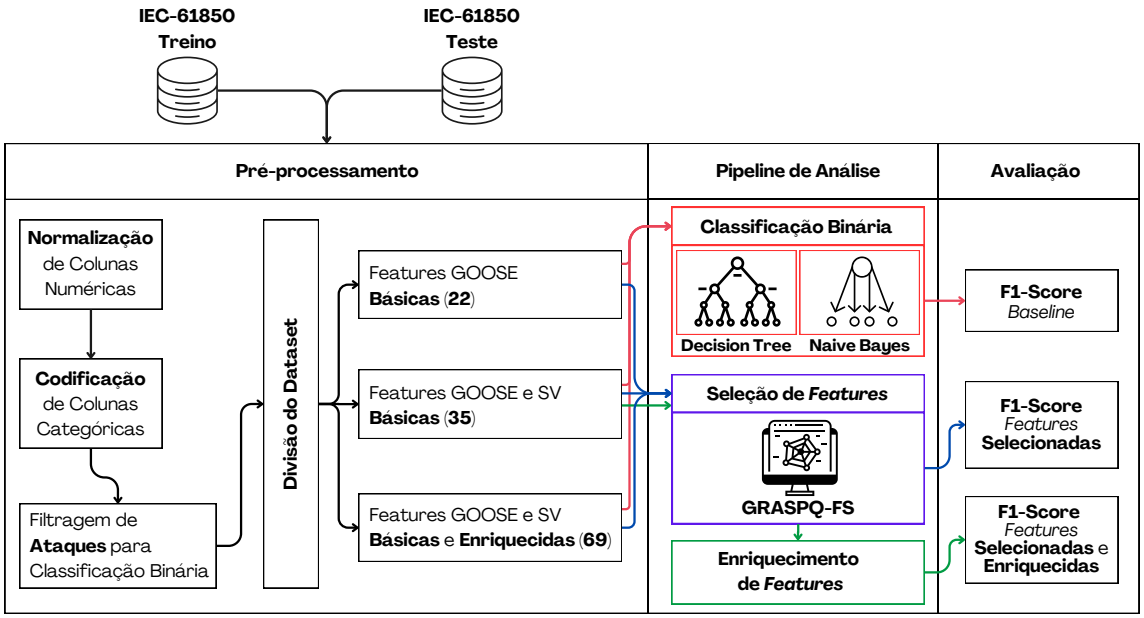


Figura 3. Fluxo Experimental para Avaliação do Desempenho do IDS no *Dataset* IEC-61850.

No pipeline de análise, primeiro realizamos a classificação binária para os três primeiros conjuntos de *features*: **1)** GOOSE básicas, 22 *features*, **2)** GOOSE e SV básicas combinadas, 35 *features*, e **3)** GOOSE e SV básicas enriquecidas, 69 *features*, utilizando algoritmos padronizados, que no nosso caso são *Decision Tree* e *Naive Bayes*. Essa avaliação gerou um F1-Score inicial que serviu como *baseline* para comparações futuras (representado pelas setas vermelhas na Figura 3). Na etapa seguinte, aplicamos a seleção de *features* utilizando o GRASPQ-FS [Quincozes et al. 2024b], que gerou mais três conjuntos de *features*: **4)** GRASPQ-FS aplicado em GOOSE básicas, 5 *features*, **5)** GRASPQ-FS aplicado em GOOSE e SV básicas combinadas, 5 *features*, e **6)** GRASPQ-FS aplicado em GOOSE e SV básicas enriquecidas, 5 features. Foram avaliadas filas de prioridade de tamanhos 10 e 100, gerando, respectivamente, 1.000 e 10.000 iterações de busca local, com base em 100 iterações na fase de construção [Quincozes et al. 2024b]. O F1-Score obtido após a seleção foi utilizado para avaliar os ganhos proporcionados por essa redução (representado pelas setas azuis na Figura 3). Por fim, avaliamos o impacto da combinação entre seleção e enriquecimento de *features*, gerando o último conjunto de *features*, **7**) onde as *features* resultantes da seleção no conjunto GOOSE e SV básicas foram enriquecidas, resultando em um número variável de *features* (representado pelas setas verdes na Figura 3).

A avaliação final dos resultados foi conduzida com base no F1-Score, calculado para os três cenários experimentais: classificação binária inicial, após a seleção de *features* e após o enriquecimento das *features* selecionadas. Esse fluxo experimental permitiu analisar detalhadamente como as técnicas de seleção e enriquecimento contribuem, individual e conjuntamente, para melhorar o desempenho do IDS em cenários complexos.

5. Resultados e Discussões

Inicialmente, foi avaliado o impacto do tamanho da fila de prioridade do GRASPQ-FS (Subseção 5.1) no *dataset* IEC-61850. Em seguida, foram analisados os efeitos da seleção

e do enriquecimento de *features* (Subseção 5.2). Por fim, foi investigado o impacto da ordem de execução desses dois processos (Subseção 5.3).

5.1. Avaliação do Tamanho da Fila de Prioridade no GRASPQ-FS

Para avaliar a eficiência do GRASPQ-FS no *dataset* IEC-61850, foram realizados experimentos com o classificador *Naive Bayes*, comparando filas de prioridade (PQ) de tamanhos 10 e 100. Estudos anteriores já indicavam que filas menores reduzem o custo computacional com impacto mínimo no F1-Score [Quincozes et al. 2024b]. Contudo, devido à complexidade do *dataset*, foi necessário verificar essa hipótese em nossos cenários.

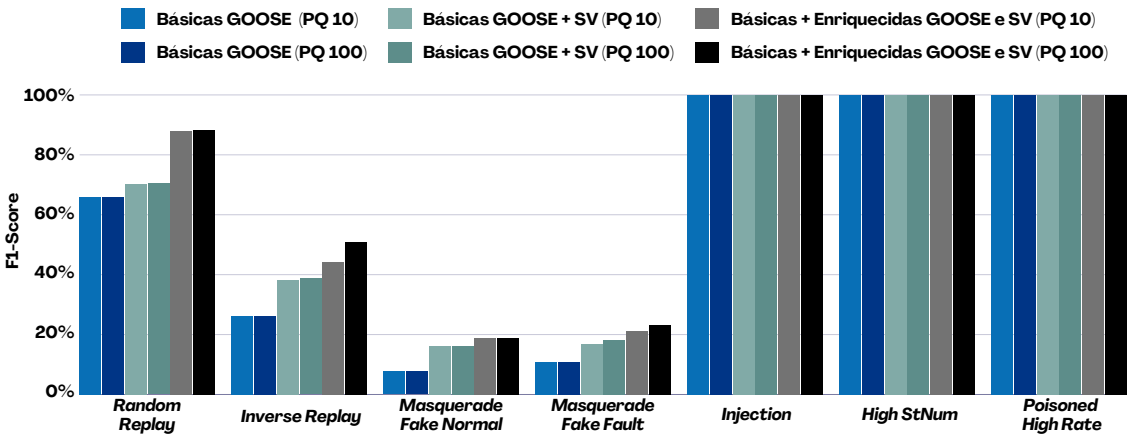


Figura 4. Comparação do F1-Score do GRASPQ-FS com Fila de Prioridade de Tamanho 10 versus 100.

A Figura 4 apresenta os F1-Scores para cada classe do *dataset* e para os diferentes conjuntos de *features*. Os resultados indicam que a diferença de desempenho entre as filas de tamanho 10 e 100 é mínima em todos os ataques. Em casos como *Injection*, *Random Replay*, *High StNum* e *Poisoned High Rate*, os F1-Scores foram praticamente idênticos, independentemente do conjunto de *features*. Já os baixos desempenhos em ataques da classe *Masquerade* refletem limitações do próprio classificador *Naive Bayes* em capturar padrões complexos, e não da estratégia de seleção ou do tamanho da fila.

A Figura 5 mostra que, enquanto o tempo com fila de prioridade 10 permaneceu abaixo de 2.000 segundos para todas as classes, a configuração com fila 100 variou entre 10.000 e 16.000 segundos. Essa diferença significativa no custo computacional reforça a eficiência da configuração com fila menor, especialmente em contextos de alta dimensionalidade como o IEC-61850. Por esse motivo, os experimentos subsequentes adotam a fila de tamanho 10, buscando um equilíbrio entre desempenho e eficiência.

5.2. Avaliação do Impacto da Seleção e Enriquecimento de *Features*

Os resultados obtidos com seis diferentes configurações de *features* são apresentados a seguir. Inicialmente, avaliamos os três conjuntos de *features*: (i) básicas do GOOSE (22 *features*), (ii) a combinação das *features* básicas do GOOSE e SV (35 *features*), e (iii) o conjunto completo, que inclui as *features* básicas do GOOSE e SV, além das enriquecidas (69 *features*), gerando os resultados *baseline*. Em seguida, é aplicada a seleção de *features* a cada um desses conjuntos. Assim, é possível a avaliação do impacto do enriquecimento das *features* básicas GOOSE e SV e da seleção de *features* em todos os três conjuntos.

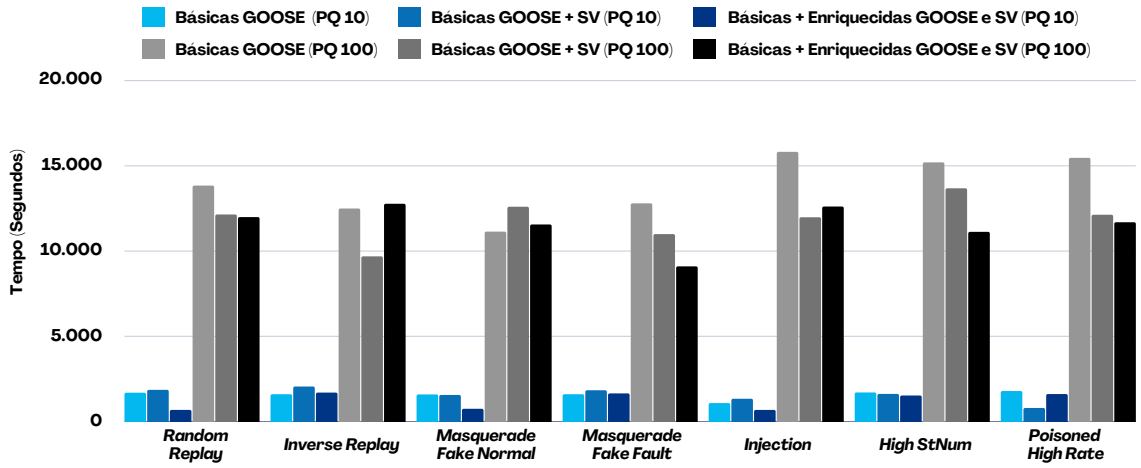


Figura 5. Comparação do Tempo de Execução do GRASPQ-FS com Fila de Prioridade de Tamanho 10 versus 100.

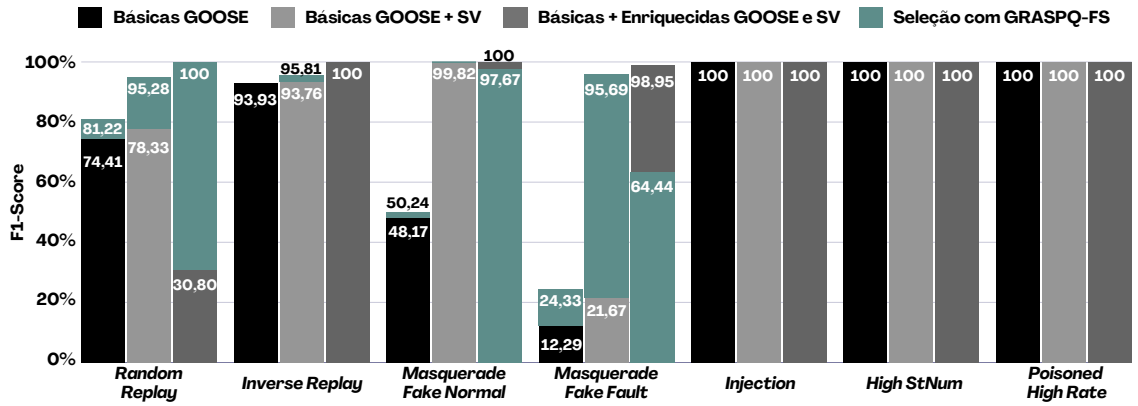


Figura 6. F1-Score do algoritmo Decision Tree.

A Figura 6 ilustra os resultados para o algoritmo *Decision Tree*. Os resultados destacam o impacto positivo do enriquecimento de *features* em comparação aos conjuntos básicos. Por exemplo, para o ataque *Masquerade Fake Fault*, o F1-Score subiu de 12,29% (básicas do GOOSE) para 21,67% (com *features* básicas do GOOSE e SV) e finalmente para 98,95% (com o enriquecimento das *features* básicas do GOOSE e SV), evidenciando a capacidade das *features* enriquecidas em capturar padrões mais representativos em cenários complexos. Para ataques como *Masquerade Fake Normal*, o conjunto combinado de *features* básicas do GOOSE e SV já apresentou um ganho significativo em relação às *features* básicas do GOOSE (48,17% para 99,82%), mas o enriquecimento elevou o desempenho para 100%.

Além disso, a aplicação do GRASPQ-FS para seleção de *features* apresentou melhorias em vários cenários, como ilustrado pelas barras verdes no gráfico. Por exemplo, no caso de *Random Replay*, o F1-Score para o conjunto enriquecido passou de 30,80% para 100% após a seleção, demonstrando a eficácia da técnica em refinar *features* relevantes em tal cenário. Outro destaque é o ataque *Masquerade Fake Fault*, onde a seleção aumentou o desempenho das *features* básicas do GOOSE e SV de 21,67% para 95,69%, um ganho significativo de mais de 70%. No entanto, nem todos os cenários demonstra-

ram melhorias consistentes. Para o ataque *Masquerade Fake Fault*, a seleção de *features* no conjunto enriquecido reduziu o F1-Score de 98,85% para 64,44%, representando uma queda de 34,51%. De maneira similar, no conjunto enriquecido, o F1-Score diminiu de 100% para 97,67% no ataque *Masquerade Fake Normal*. Esses resultados sugerem que, em alguns casos, a seleção pode excluir *features* que capturam padrões importantes, impactando negativamente o desempenho em ataques mais complexos — algo que será explorado em mais detalhes na análise de explicabilidade na Seção 6.

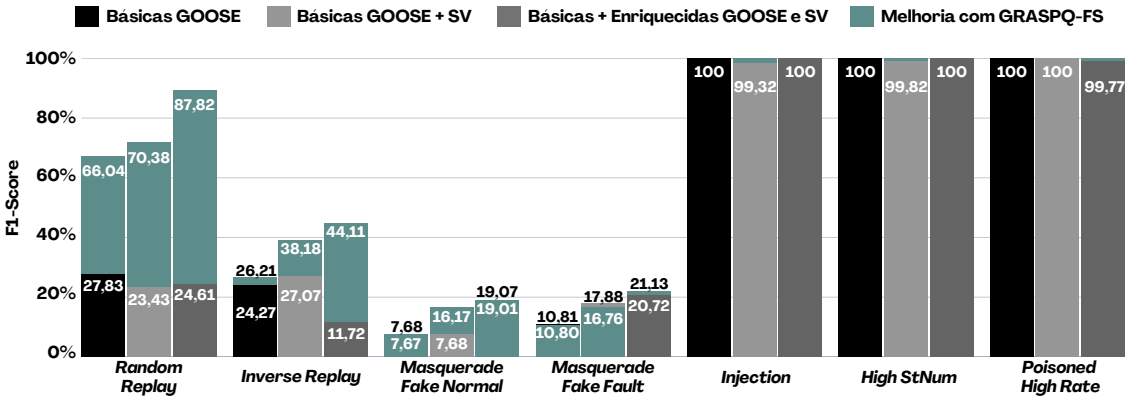


Figura 7. F1-Score do algoritmo Naive Bayes.

Os resultados do algoritmo *Naive Bayes* são apresentados na Figura 7. Tal algoritmo evidencia um padrão diferente de comportamento em relação ao *Decision Tree*. De maneira geral, o desempenho do *Naive Bayes* foi inferior para diversos ataques. Para o ataque *Random Replay*, o F1-Score foi semelhante entre os três conjuntos avaliados antes da seleção de *features*: 27,83% para as *features* básicas do GOOSE, 23,43% para a combinação GOOSE + SV, e 24,61% para o conjunto enriquecido. No entanto, a aplicação do GRASPQ-FS trouxe uma melhoria significativa, elevando o desempenho para 66,04% (GOOSE básicas), 70,38% (GOOSE + SV básicas) e 87,82% (enriquecidas). Esses resultados demonstram que, embora o enriquecimento sozinho não tenha gerado ganhos substanciais, a seleção de *features* conseguiu capturar e priorizar as *features* mais relevantes, potencializando o desempenho significativamente.

Outro destaque é a classe *Inverse Replay*, que apresentou F1-Scores antes da seleção de *features* de 24,27% para as *features* básicas do GOOSE, 27,07% para a combinação GOOSE + SV, e 11,72% para o conjunto enriquecido. Esse resultado indica que o enriquecimento, nesse caso, introduziu *features* que não contribuíram para o modelo, possivelmente gerando ruídoa. No entanto, a aplicação do GRASPQ-FS foi capaz de refinar as *features* mais relevantes, resultando em melhorias expressivas: o F1-Score subiu para 26,21% (GOOSE básicas), 38,18% (GOOSE + SV básicas) e 44,11% (enriquecidas). Esses resultados demonstram que a seleção de *features* foi fundamental para recuperar e aprimorar o desempenho do modelo, mesmo em um cenário onde o enriquecimento isolado teve impacto negativo.

Já em cenários mais complexos, como os ataques *Masquerade Fake Normal* e *Masquerade Fake Fault*, o impacto do enriquecimento foi limitado, com ganhos modestos nos F1-Scores: de 7,68% para 19,07% em *Masquerade Fake Normal* e de 10,81% para 20,72% em *Masquerade Fake Fault*. No entanto, mesmo após a aplicação do GRASPQ-

FS, os valores permaneceram baixos, indicando que o *Naive Bayes* encontra maior dificuldade em lidar com padrões avançados representados nesses ataques. Esses resultados reforçam que, embora o *Naive Bayes* possa ser eficaz em cenários mais simples, sua aplicabilidade é limitada em ataques mais sofisticados, como os da classe *Masquerade*.

Os demais ataques, como *Injection*, *High StNum* e *Poisoned High Rate*, apresentaram desempenhos elevados tanto para o *Naive Bayes* quanto para o *Decision Tree*, independentemente do conjunto de *features* avaliado. Esses resultados indicam que a natureza desses ataques facilita sua identificação, possivelmente devido a padrões mais claros e comportamentos menos complexos, que são capturados de forma eficiente mesmo por modelos mais simples e em cenários sem enriquecimento ou seleção de *features*. Esse desempenho robusto sugere que ataques com características bem definidas são menos dependentes de técnicas avançadas de pré-processamento, destacando a necessidade de concentrar esforços em ataques mais sofisticados, como os das classes *Masquerade*, onde os resultados ainda revelam desafios significativos.

5.3. Análise do Impacto da Ordem: Seleção antes do Enriquecimento de *Features*

Nesta subseção, avaliamos o impacto da ordem de operação entre seleção e enriquecimento de *features*. A análise considera dois cenários principais: (i) aplicação do GRASPQ-FS após o enriquecimento (como considerado acima), ou seja, a seleção aplicada às *features* básicas e enriquecidas do GOOSE e SV, e (ii) enriquecimento das *features* após a seleção, ou seja, o GRASPQ-FS é aplicado ao conjunto básico GOOSE + SV e, em seguida, as *features* selecionadas são enriquecidas, caso aplicável, conforme definido na Figura 2. Por exemplo, suponha que o GRASPQ-FS tenha retornado cinco *features*: `stNum`, `sqNum`, `ethDst`, `isbA` e `time`. Três novas *features* podem ser derivadas dessas (veja Figura 2), e contribuiriam para um novo conjunto de oito *features*: `stNum`, `stDiff` (enriquecida de `stNum`), `sqNum`, `sqDiff` (enriquecida de `sqNum`), `ethDst`, `isbA`, `isbARmsValue` e `isbATrapAreaSum` (enriquecidas de `isbA`) e `time`.

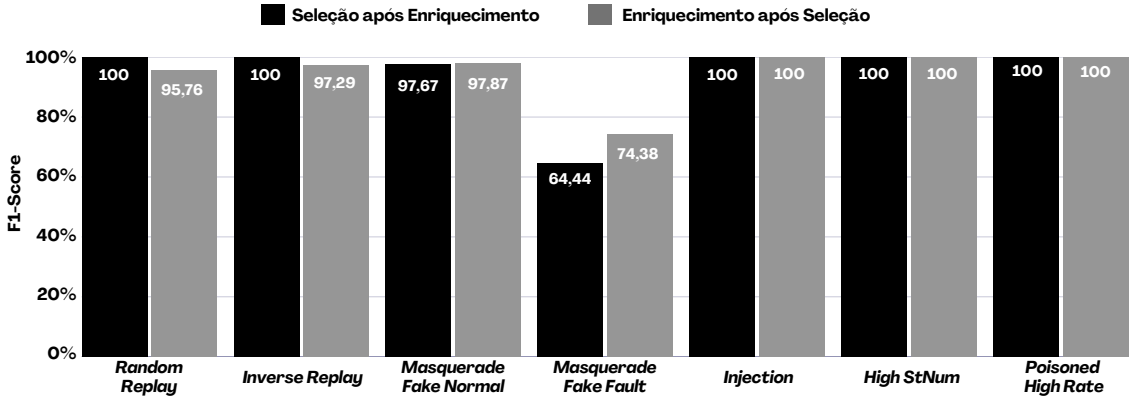


Figura 8. Comparação entre Seleção de *Features* e Seleção seguida de Enriquecimento nas *Features* Básicas GOOSE + SV, com o *Decision Tree*.

A Figura 8 apresenta a comparação entre o desempenho obtido com as *features* retornadas pelo GRASPQ-FS (representadas pelas barras pretas) e o desempenho após o enriquecimento (representadas pelas barras cinzas), utilizando o algoritmo *Decision Tree*. Para o ataque *Random Replay*, selecionar *features* após o enriquecimento resultou em

ganho de desempenho, de 95,76% para 100%. De maneira similar, no caso do *Inverse Replay*, o F1-Score subiu de 97,29% para 100%, sugerindo que primeiro enriquecer e depois selecionar *features* é mais eficiente para capturar os padrões desses ataques.

Por outro lado, enriquecer as *features* previamente selecionadas demonstrou ser vantajoso em cenários mais complexos, como os ataques *Masquerade Fake Normal* e *Masquerade Fake Fault*. Para o ataque *Masquerade Fake Normal*, o F1-Score aumentou de 97,67% para 97,87%, representando um ganho de 0,20%. Já para o ataque *Masquerade Fake Fault*, o ganho foi mais expressivo, com um aumento de 64,44% para 74,38%, ou seja, 9,94%. Esses resultados sugerem que o enriquecimento pode complementar as *features* selecionadas em cenários onde a complexidade dos padrões requer informações adicionais para serem plenamente representadas, destacando a importância de avaliar a combinação dessas técnicas de forma contextualizada.

A Figura 9 compara duas abordagens para combinar seleção e enriquecimento de *features*: a seleção após o enriquecimento (barras pretas) e o enriquecimento após a seleção (barras cinzas), utilizando o *Naive Bayes*. Esses resultados destacam diferenças significativas entre as estratégias, variando de acordo com a natureza de cada ataque.

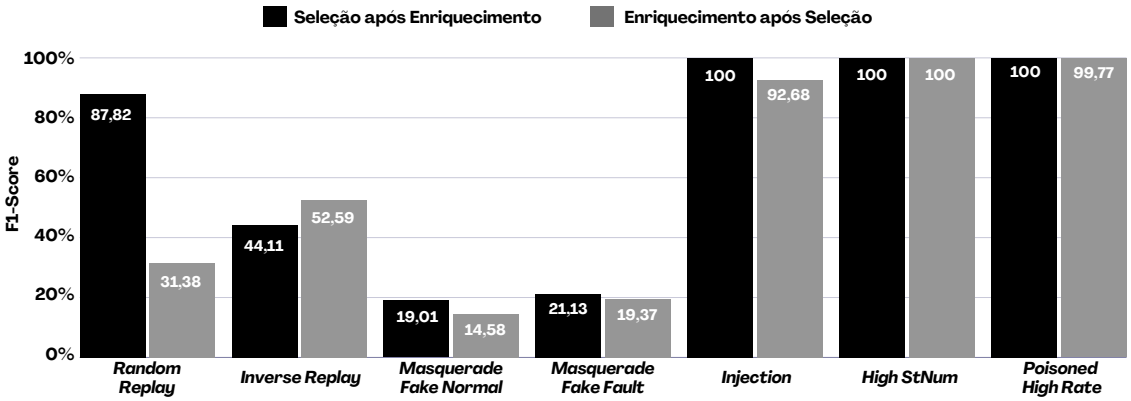


Figura 9. Comparação entre Seleção de *Features* e Seleção seguida de Enriquecimento nas *Features* Básicas GOOSE + SV, com o *Naive Bayes*.

Para a classe *Random Replay*, a seleção após o enriquecimento produziu o melhor desempenho, alcançando um F1-Score de 87,82%, enquanto o enriquecimento após a seleção apresentou um resultado inferior de 31,38%. Essa diferença indica que as *features* enriquecidas no conjunto completo capturam melhor os padrões desse ataque e, quando combinadas com a seleção posterior, resultam em um desempenho mais robusto.

No ataque *Inverse Replay*, o enriquecimento após a seleção foi a abordagem que apresentou ganhos, com um aumento no F1-Score de 44,11% (seleção após enriquecimento) para 52,59%. Esse cenário sugere que algumas *features* enriquecidas adicionadas ao conjunto reduzido de *features* selecionadas pelo GRASPQ-FS contribuíram para capturar informações relevantes ao ataque.

Por outro lado, ataques como *Masquerade Fake Normal* e *Masquerade Fake Fault* apresentaram resultados mais consistentes com a seleção após o enriquecimento. No *Masquerade Fake Normal*, a seleção após enriquecimento resultou em um F1-Score de 19,01%, contra 14,58% do enriquecimento após a seleção. No entanto, no *Masquerade*

Fake Fault, a diferença foi menor, com 21,13% para a seleção após enriquecimento, enquanto o enriquecimento após a seleção apresentou 19,37%.

Para os ataques *Injection*, *High StNum* e *Poisoned High Rate*, ambas as abordagens resultaram em F1-Scores próximos de 100%, indicando que a natureza desses ataques é mais diretamente representada pelas *features* básicas e enriquecidas, tornando a interação entre seleção e enriquecimento menos influente.

Portanto, para cinco dos sete cenários avaliados, a abordagem de realizar a seleção após o enriquecimento foi mais eficaz. No entanto, os ganhos no ataque *Inverse Replay* sugerem que, em alguns casos, enriquecer um subconjunto reduzido de *features* pode ajudar a capturar padrões específicos que o conjunto completo não reflete com precisão. Esses achados reforçam que a escolha entre essas abordagens deve considerar a natureza específica dos ataques e as características das *features* disponíveis.

6. Explicabilidade dos Resultados Fora dos Padrões

Conforme discutido nas seções anteriores, alguns resultados observados não seguiram o padrão esperado. Um exemplo é o desempenho do algoritmo *Decision Tree* no ataque *Masquerade Fake Fault*, onde o F1-Score foi reduzido de 98,95% (com o conjunto completo de *features*) para 64,44% (após a seleção das *features* utilizando GRASPQ-FS), como ilustrado na Figura 6. Além disso, foi identificado que, nesse caso específico, a abordagem de enriquecimento das *features* após a seleção proporcionou uma melhoria no F1-Score, que passou de 64,44% para 74,38% (Figura 8).

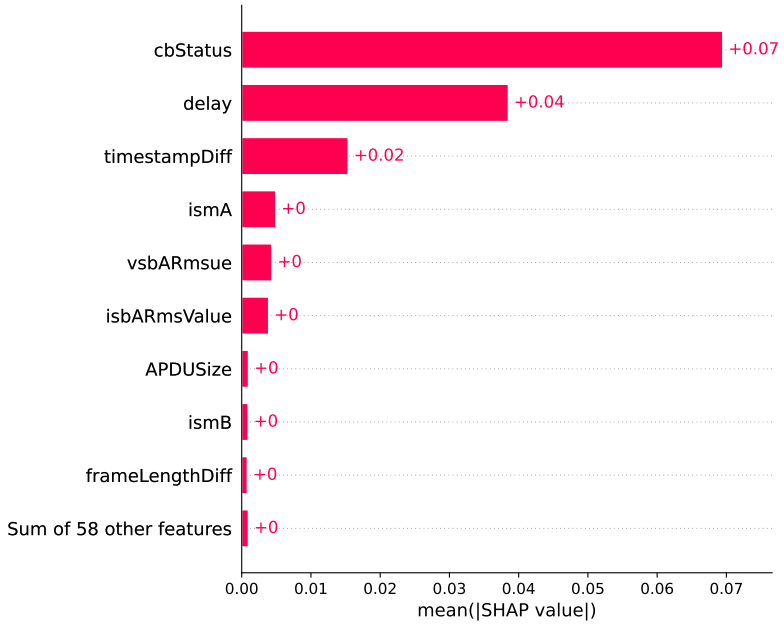


Figura 10. Técnica SHAP aplicada ao Maquerade Fake Fault.

Para entender melhor esses resultados atípicos, utilizamos técnicas de *eXplainability Artificial Intelligence* (XAI) [Bragança et al. 2023], mais especificamente o método *SHapley Additive exPlanations* (SHAP) [Lundberg and Lee 2017], que permite a explicação das saídas de modelos de aprendizado de máquina de forma interpretável.

A Figura 10 apresenta os resultados do SHAP para o *Masquerade Fake Fault*. É importante notar que a F1-Score de 64,44% foi obtida através da seleção das seguintes *features* pelo o GRASPQ-FS: `cbStatus`, `timestampDiff`, `vsbARmsue`, `ismCRmsValue` e `protocol_GOOSE`. Contudo, conforme mostrado na Figura 10, as *features* mais influentes para a detecção do Masquerade Fake Fault são `cbStatus`, `delay` e `timestampDiff`, com valores médios de SHAP superiores a 0,01. Essas *features* seriam, portanto, as mais relevantes e deveriam estar entre as selecionadas pelo GRASPQ-FS. No entanto, a *feature* `delay` não foi incluída, o que explica, em parte, o baixo desempenho observado.

Para compreender o motivo de o GRASPQ-FS não ter selecionado a *feature* `delay`, investigamos o processo de seleção do algoritmo. O primeiro passo do GRASPQ-FS envolve a construção da *Restricted Candidate List* (RCL) utilizando o algoritmo de ranqueamento por *Mutual Information* (MI). Detalhes sobre o funcionamento do GRASPQ-FS podem ser encontrados em [Quincozes et al. 2024b]. Em nossos experimentos, a RCL consistiu de 25 *features*. No entanto, o ranqueamento baseado em MI colocou a *feature* `delay` na 38ª posição, o que fez com que ela não fosse incluída na RCL e, consequentemente, não avançasse para as fases subsequentes do GRASPQ-FS. Isso evidencia uma limitação do uso de MI, uma técnica univariada que ignora interações entre features [Nayak et al. 2025], o que pode levar à subvalorização de variáveis que só se mostram relevantes em conjunto com outras, especialmente em padrões temporais complexos, como nos ataques *Masquerade*.

Por fim, experimentos adicionais mostraram que, ao utilizar apenas as 5 principais *features*, o F1-Score alcançou 99,95%, confirmando que a *feature* `delay` tem um alto impacto no modelo, especialmente em ataques *Masquerade Fake Fault*. Esse resultado reforça que aplicar o enriquecimento antes da seleção é crucial para garantir que *features* derivadas altamente relevantes — *e.g.,* como a `delay`, obtida a partir da combinação entre `gTimestamp` (GOOSE) e `time` (SV) — estejam disponíveis para avaliação.

7. Conclusão

Este trabalho apresentou uma avaliação extensiva do impacto da seleção e do enriquecimento de *features* no desempenho de IDSs aplicados a sistemas ciberfísicos, com foco no *dataset* IEC-61850. Os experimentos exploraram três conjuntos principais de *features* (básicas GOOSE, básicas GOOSE + SV e GOOSE + SV enriquecidas), avaliando o efeito isolado e combinado das técnicas em sete ataques distintos.

Os resultados mostraram que a seleção com o GRASPQ-FS foi eficaz em ataques como *Random Replay*, elevando o F1-Score de 30,8% para 100%, e também trouxe ganhos em cenários complexos como *Masquerade Fake Fault*, com aumento de 21,7% para 95,69%. O enriquecimento, por sua vez, foi decisivo em ataques mais sofisticados, elevando o F1-Score do mesmo *Masquerade Fake Fault* de 21,7% para 98,95%. A combinação das duas técnicas proporcionou os melhores resultados, embora com sensibilidade à ordem de aplicação: aplicar a seleção diretamente sobre o conjunto enriquecido pode eliminar *features* críticas quando se utiliza ranqueamento univariado. A análise de explicabilidade com SHAP confirmou esse risco, ao mostrar que *features* derivadas altamente relevantes (*i.e.*, `delay`) podem ser descartadas se o enriquecimento não for realizado previamente. Em experimentos adicionais, o uso apenas das cinco *features* com

maior impacto SHAP elevou o F1-Score para 99,95%, reforçando que enriquecer antes da seleção é essencial para garantir a disponibilidade dessas variáveis.

Como trabalhos futuros, pretende-se incorporar o SHAP ao processo do GRASPQ-FS, substituindo o ranqueamento baseado em MI por uma abordagem orientada por explicabilidade. Também planeja-se utilizar outros *datasets* com *features* enriquecidas e explorar algoritmos alternativos de seleção.

Agradecimentos

Este trabalho foi realizado com o apoio da CAPES, CNPq, FAPERJ e do Departamento de Energia dos Estados Unidos, sob o Número de Concessão [DE-CR0000039].

Referências

Bragança, H., Rocha, V., Souto, E., Kreutz, D., and Feitosa, E. (2023). Explaining the effectiveness of machine learning in malware detection: Insights from explainable ai. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 181–194. SBC.

Horchulhack, P., Viegas, E. K., Santin, A. O., and Geremias, J. (2022). Atualizaçao de modelo baseado em aumento de dados e transferência de aprendizagem para detecçao de intrusao em redes. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 223–235. SBC.

Kaushik, B., Sharma, R., Dhama, K., Chadha, A., and Sharma, S. (2023). Performance evaluation of learning models for intrusion detection system using feature selection. *Journal of Computer Virology and Hacking Techniques*, 19(4):529–548.

Latif, N., Ma, W., and Ahmad, H. B. (2025). Advancements in securing federated learning with ids: a comprehensive review of neural networks and feature engineering techniques for malicious client detection. *Artificial Intelligence Review*, 58(3):91.

Li, J., Othman, M. S., Chen, H., and Yusuf, L. M. (2024). Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. *Journal of Big Data*, 11(1):36.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Mackiewicz, R. E. (2006). Overview of IEC 61850 and Benefits. In *2006 IEEE Power Engineering Society General Meeting*, pages 8–pp. IEEE.

Mariani, W. C., Munaretto, A., Fonseca, M., Lopes, H., and Silva, T. H. (2024). Detecção de intrusão e análise cyberfísica em redes industriais. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 787–793. SBC.

Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., and Mohammad, R. M. (2023). Intrusion detection system using feature extraction with machine learning algorithms in IoT. *Journal of Sensor and Actuator Networks*, 12(2):29.

Nayak, G. S., Muniyal, B., and Belavagi, M. C. (2025). Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models. *IEEE Access*, 13:33308–33320.

Ngo, V.-D., Vuong, T.-C., Van Luong, T., and Tran, H. (2024). Machine learning-based intrusion detection: feature selection versus feature extraction. *Cluster Computing*, 27(3):2365–2379.

Quincozes, S. E., Albuquerque, C., Passos, D., and Mossé, D. (2024a). ERENO: A Framework for Generating Realistic IEC–61850 Intrusion Detection Datasets for Smart Grids. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3851–3865.

Quincozes, V. E., Quincozes, S. E., Albuquerque, C., Passos, D., and Massé, D. (2024b). Efficient feature selection for intrusion detection systems with priority queue-based GRASP. In *International Conf. on Cloud Networking (CloudNet)*, pages 1–8. IEEE.

Quincozes, V. E., Quincozes, S. E., Passos, D., Albuquerque, C., and Mossé, D. (2024c). Towards feature engineering for intrusion detection in IEC–61850 communication networks. *Annals of Telecommunications*, pages 1–15.

Rabie, O. B. J., Balachandran, P. K., Khojah, M., and Selvarajan, S. (2022). A proficient ZESO-DRKFC model for smart grid SCADA security. *Electronics*, 11(24):4144.

Rahim, S. A. and Manoharan, A. (2024). An Optimization-based Feature Selection and Hybrid Spiking VGG 16 for Intrusion Detection in the CPS Perception layer. *IEEE Access*.

Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., and Portmann, M. (2024). Feature extraction for machine learning-based intrusion detection in IoT networks. *Digital Communications and Networks*, 10(1):205–216.

Technologies, C. P. S. (2024). Cyber Security Report 2024. Disponível em: `https://www.checkpoint.com/resources/report-3854/report--cyber-security-report-2024`. Acessado em: Janeiro de 2025.

Thakkar, A. and Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, 55(1):453–563.

Xi, C., Wang, H., and Wang, X. (2024). A novel multi-scale network intrusion detection model with transformer. *Scientific Reports*, 14(1):23239.

Zouhri, H., Idri, A., and Ratnani, A. (2024). Evaluating the impact of filter-based feature selection in intrusion detection systems. *International Journal of Information Security*, 23(2):759–785.