Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.6186

IWSHAP-X: Enhancing Feature Selection for Intrusion Detection Systems via XAI-Guided Metaheuristics

Felipe H. Scherer [**Federal University of Pampa** | ***felipescherer.aluno@unipampa.edu.br***]
Felipe N. Dresch [**Federal University of Pampa** | ***felipedresch.aluno@unipampa.edu.br***]
Matheus M. Ciocca [**Federal University of Pampa** | ***matheusciocca.aluno@unipampa.edu.br***]
Silvio E. Quincozes [**Federal University of Pampa, Federal University of Uberlândia** | ***silvioquincozes@unipampa.edu.br***]
Diego Kreutz [**Federal University of Pampa** | ***diegokreutz@unipampa.edu.br***]
Vagner E. Quincozes [**Fluminense Federal University** | ***vequincozes@midiacom.uff.br***]

AI Horizon Labs, Federal University of Pampa. Ave. Tiaraju 810, Alegrete, RS, 97546-550, Brazil.

Received: 29 June 2025 • **Accepted:** 31 October 2025 • **Published:** 07 May 2026

Abstract Feature selection plays a key role in developing effective machine learning-based Intrusion Detection Systems (IDS), as it influences model performance, computational efficiency, and explainability. While traditional methods like filter, wrapper, and embedding approaches have shown value, they frequently encounter challenges with premature convergence that can result in less optimal feature subsets. We present IWSHAP-X (IWSHAP with eXploration), an enhanced hybrid approach that combines SHapley Additive Explanations (SHAP) feature importance rankings with metaheuristic search strategies. This method extends the original IWSHAP process by introducing an additional exploration phase designed to reduce the likelihood of converging to local optima during feature selection. Our experiments with IWSHAP-X on the X-CANIDS dataset across multiple attack scenarios reveal several advantages over the original IWSHAP method. The approach demonstrates improved feature reduction capabilities while maintaining classification accuracy, along with better computational efficiency. Specifically, IWSHAP-X achieves up to 53.13% fewer selected features compared to IWSHAP, without compromising classification performance. These results suggest that IWSHAP-X offers a viable solution for IDS applications where both feature reduction and model effectiveness are important considerations.

Keywords: Machine Learning, Feature Selection, Metaheuristics, Premature Convergence, Explainable Artificial Intelligence (XAI)

1 Introduction

The increasing sophistication and volume of cyberattacks targeting critical infrastructure and networked systems require robust and efficient Intrusion Detection Systems (IDS). Machine Learning-based IDSs offer a promising solution, as they can identify malicious activities by learning patterns from network traffic [Bari *et al*., 2023; Santhosh Kumar *et al*., 2023; Buiya *et al*., 2024; Rana *et al*., 2024; Pasupathi *et al*., 2025; Mohamed, 2025]. However, the effectiveness of these systems depends heavily on the quality and relevance of the features used to train ML models. Feature selection is crucial for optimizing IDS performance, lowering computational costs, and enhancing detection accuracy [Halim *et al*., 2021; Xie *et al*., 2023; Li *et al*., 2024; Mallidi and Ramisetty, 2025; Madhloom Kurdi *et al*., 2025; Salehpour *et al*., 2025].

Feature selection techniques are usually divided into filter, wrapper, and embedded methods. Filter methods are computationally efficient but may miss complex feature interactions. Wrapper methods evaluate subsets based on model performance, capturing interactions more effectively, but they can be computationally expensive, especially in real-time scenarios. Embedded methods integrate feature selection into the learning process, making them more efficient than wrapper methods, though they are often limited to specific algorithms [Chandrashekar and Sahin, 2014; Pudjihartono *et al*., 2022; Theng and Bhoyar, 2024; Song *et al*., 2024].

Recent advances in eXplainable Artificial Intelligence (XAI), such as SHapley Additive exPlanations (SHAP), have improved feature selection by providing interpretable insights into feature importance [Khani *et al*., 2024; Van Zyl *et al*., 2024; Santos *et al*., 2024; Singh *et al*., 2025; Fatema *et al*., 2025]. SHAP values quantify the contribution of each feature to a model's predictions, enabling more informed and transparent feature selection. This approach helps prioritize the most relevant features, potentially improving model performance while maintaining interpretability.

In our previous work, IWSHAP [Scherer *et al*., 2024], we combined the Incremental Wrapper Subset Selection (IWSS) algorithm [Bermejo *et al*., 2009] with SHAP values to enhance feature selection. This method balanced model performance and computational efficiency by leveraging SHAP-based feature rankings within a wrapper framework. However, IWSHAP's reliance on the greedy IWSS algorithm sometimes

led to suboptimal feature subsets, as it could discard potentially useful features prematurely, limiting further exploration of feature combinations.

To address this limitation, we introduce IWSHAP-X, an improved feature selection approach that integrates metaheuristic optimization with XAI-driven feature ranking. Unlike IWSS, which may converge to local optima, metaheuristics employ stochastic search strategies to explore a broader range of feature subsets. This enhances the balance between model performance, feature subset size, and computational efficiency. By combining metaheuristic principles with SHAP values and greedy selection, IWSHAP-X achieves better feature selection outcomes.

We evaluate IWSHAP-X on the X-CANIDS dataset [Jeong *et al*., 2024a], which focuses on Control Area Networks (CAN) used for intra-vehicular communication. Our results show that IWSHAP-X reduces the number of selected features by up to 53.13% while maintaining or improving performance metrics such as F1-Score compared to IWSHAP. Additionally, it significantly lowers computational costs, making it suitable for real-time and resource-constrained environments.

The rest of this paper is organized as follows: Section 2 provides an overview of metaheuristic optimization and XAI, Section 3 reviews related work, and Section 4 describes the proposed algorithm step by step. Section 5 presents the experimental setup and results, and Section 6 concludes the paper with future research directions.

2 Background

We begin by presenting the fundamental concepts and techniques that support our research. In the subsequent subsections, we cover four key areas: IDS, feature selection methods, metaheuristic optimization strategies, and explainable AI (XAI). For each area, we explain how it contributes to the development of our feature selection approach.

2.1 Intrusion Detection Systems

We recognize IDSs as vital elements in modern cybersecurity infrastructures. These systems monitor and analyze network or system activities to detect malicious behavior or security policy violations [Mitchell and Chen, 2014]. They provide a crucial additional defense layer that can identify threats potentially missed by traditional security measures like firewalls or antivirus software. IDSs typically fall into two main categories: signature-based and anomaly-based detection.

Signature-based IDSs work by matching observed traffic or behavior against a database of known attack patterns (called "*signatures*"). These signatures act as unique identifiers for documented threats. When the system finds a match, it generates an alert. This approach offers high accuracy for known attacks and maintains low false positive rates. However, we note its significant limitation: the inability to detect new or zero-day attacks lacking predefined signatures [Axelsson, 2000; Azam *et al*., 2023; Abdulganiyu *et al*., 2024].

Anomaly-based IDSs take a different approach. They first establish models of normal system behavior using statistical profiles, heuristic rules, or ML techniques. The systems then flag substantial deviations from this baseline as potential threats [Quincozes *et al*., 2020]. This method proves particularly valuable for detecting novel attacks. Still, we observe its main drawback: the tendency to generate false positives by misclassifying legitimate but unusual activities as malicious. This can undermine system trustworthiness and increase analyst workload.

We find that regardless of the detection method, IDS performance depends heavily on feature quality and relevance. Inappropriate feature selection can significantly impair system accuracy and efficiency, potentially causing both missed detections and false alarms.

Consequently, we emphasize that proper feature selection becomes vital for:

- Improving an IDS's discrimination between normal and malicious activities;
- Optimizing resource utilization;
- Enhancing the overall robustness of intrusion detection.

Ultimately, we conclude that successful intrusion detection requires not only effective algorithms but also careful consideration of input data selection and preprocessing.

2.2 Feature Selection

We examine feature selection as a process to identify the most relevant features from a larger set, which helps reduce computational requirements, enhance model performance, and improve result interpretability [Li *et al*., 2017; Pudjihartono *et al*., 2022; Salehpour *et al*., 2025]. The field typically organizes feature selection methods into three categories: filter, wrapper, and embedded approaches [Chandrashekar and Sahin, 2014; Pudjihartono *et al*., 2022; Lamsaf *et al*., 2025].

Filter methods operate by evaluating individual features using statistical measures or information criteria without involving a learning algorithm [Esseghir, 2010; Cai *et al*., 2018]. While we find these methods computationally efficient, we note they might overlook important relationships between features. Wrapper methods address this limitation by assessing feature subsets through repeated model training and evaluation, providing a more complete picture of feature importance [Bermejo *et al*., 2009; Cai *et al*., 2018]. However, we recognize this comes at the cost of increased computational demands, particularly for large datasets.

Embedded methods integrate feature selection into the learning process, as seen in decision tree algorithms that naturally select important features during model building [Guyon and Elisseeff, 2003; Chandrashekar and Sahin, 2014]. To combine the strengths of these approaches, researchers have developed hybrid methods. In our previous work, we proposed IWSHAP [Scherer *et al*., 2024], which merges SHAP-based feature importance with the Incremental Wrapper Subset Selection algorithm [Bermejo *et al*., 2009] to achieve both computational efficiency and model effectiveness.

2.3 Metaheuristic Optimization

We consider metaheuristic optimization as a family of high-level strategies for finding near-optimal solutions to complex problems. While these methods cannot guarantee finding the

global optimum, they typically provide satisfactory solutions within practical time limits. This characteristic makes them particularly well-suited for feature selection tasks, where the solution space becomes prohibitively large as the number of features increases [Rostami *et al*., 2021].

Metaheuristic algorithms share several fundamental characteristics that distinguish them from traditional optimization methods. They rely on probabilistic decision-making rather than deterministic rules, enabling broader exploration of the solution space. Their diverse exploration strategies help avoid becoming trapped in local optima, while their iterative refinement processes continuously improve solutions over time. The inherent randomness in these algorithms allows them to escape suboptimal regions while thoroughly exploring the feature space. In developing IWSHAP-X, we have incorporated these principles through probabilistic feature selection, dynamic exploration mechanisms to prevent premature convergence, and adaptive refinement processes that iteratively improve feature subsets.

The field has developed several well-established metaheuristic approaches for feature selection, each drawing inspiration from different natural phenomena. Genetic Algorithms (GA) draw from biological evolution, Particle Swarm Optimization (PSO) models the collective intelligence of swarms, and Ant Colony Optimization (ACO) replicates the foraging behavior of ants. While these methods have demonstrated effectiveness in feature selection, they share a common challenge: the risk of premature convergence. This occurs when the optimization process settles into a locally optimal solution before adequately exploring other potentially better regions of the solution space [Chaitanya *et al*., 2021].

Researchers have proposed various strategies to overcome this limitation and improve metaheuristic performance. Memory retention mechanisms help maintain diversity among candidate solutions, while backward elimination techniques systematically remove less relevant features. Search restart procedures reintroduce randomness to help the algorithm escape from local optima. These enhancements work together to preserve the algorithm's exploratory capabilities throughout the optimization process, increasing the likelihood of discovering feature subsets that improve both model performance and generalizability [Nguyen *et al*., 2014; Tran *et al*., 2014].

2.4 Explainable Artificial Intelligence

XAI seeks to clarify the decision-making processes of an AI model, making its decisions more comprehensive by pointing to which features were most or least impactful on the model's predictions [Došilović *et al*., 2018]. Among these techniques, SHAP is widely recognized for applying game theory to quantify the contribution of individual features to the model's output [Quincozes *et al*., 2024; ORG, 2024]. By providing a ranking of feature importance through SHAP Values, it is possible to infer which features are more relevant and which are less significant or even have no relevance at all (*i.e.,* lack a valid numerical representation). In certain use cases, handling these values may be necessary. Additionally, graphical representations enable a better understanding of the impact of features. By default, in SHAP plots, red values indicate that a given feature had a high impact on the prediction of an anomaly or signature, whereas blue values signify an impact on the prediction of normal samples [ORG, 2024].

3 Related Work

We review several studies that tackle feature selection challenges through diverse strategies, including memory retention, feature replacement, hybrid approaches, and metaheuristic algorithms. Table 1 summarizes these key works, focusing on their application scenarios, mitigation strategies, methodological approaches, use of metaheuristics and XAI techniques, and specific implementations in IDS contexts. In this section, we specifically examine feature selection techniques applied to IDSs, highlighting their importance for cybersecurity applications.

One persistent challenge in feature selection involves the premature elimination of informative attributes, commonly known as premature convergence or early stopping, which can compromise model quality. To address this issue, Bermejo *et al*. [2009] developed the IWSSr (Incremental Wrapper-based Subset Selection with Replacement) algorithm. This approach introduces a feature replacement mechanism that re-evaluates previously selected features during the selection process.

The IWSSr algorithm employs statistical criteria to assess both feature inclusion and removal, effectively minimizing the risk of suboptimal feature subsets. Our analysis of their results shows that IWSSr maintains the classification accuracy of the original IWSS method while reducing the number of selected features, thereby improving model efficiency. While initially applied to microarray data analysis for cancer prediction, we recognize the broader applicability of IWSSr's core strategy across different domains. The method's strength lies in its straightforward yet effective feature replacement mechanism, achieving convergence mitigation without relying on metaheuristic techniques.

Other researchers have pursued metaheuristic-based solutions. Chaitanya *et al*. [2021] examine how PSO's rapid convergence can lead to loss of information about previous good solutions. Their proposed variants - PSOMR (PSO with Memory Retention) and MS-PSOMR (Multi-swarm PSO with Memory Retention) - add memory retention mechanisms to preserve information about optimal solutions across iterations. Our analysis of their results shows these approaches successfully reduce premature convergence effects and enhance feature selection performance. However, we note these methods do not integrate hybrid approaches combining XAI techniques like SHAP-based ranking or dynamic exploration strategies to further improve feature selection.

Furthermore, we examine additional strategies that combine backward elimination with search restart mechanisms to address feature selection challenges. Nguyen *et al*. [2014] developed the PSOBE (PSO with Backward Elimination) algorithm, which employs mutual information measures to continuously reassess feature importance during optimization. This method enables more thorough exploration of the solution space while minimizing the inappropriate exclusion of relevant features. We note the algorithm's validation across diverse domains, including healthcare (Arrhyth-

Table 1. Overview of related work.
Cells Caption: ● Yes; ○ No.

Work	Scenario	Mitigation Strategy	Approach	Metaheuristics	XAI	IDS
[Bermejo *et al*., 2009]	Medical Researchs	Replacement	Wrapper	○	○	○
[Chaitanya *et al*., 2021]	Computional Optimization	Memory Retention	Hybrid	●	○	○
[Nguyen *et al*., 2014]	*Multiple*	Backward Elimination	Hybrid	●	○	○
[Tran *et al*., 2014]	Medical Researchs	Restart Mechanism	Wrapper	●	○	○
[Quincozes *et al*., 2021]	Cyber-Physical Systems	Parametrised Stop Criteria	Hybrid	●	○	●
[Vijayanand and Devaraj, 2020]	Educational Networks	Combining WOA with GA	Hybrid	●	○	●
This Work	**CAN Networks**	**Exploring Mechanism**	**Hybrid**	●	●	●

mia dataset [Guvenir and Quinlan, 1997]), image processing (Madelon dataset [Guyon, 2004]), and text analysis (Multiple Features dataset [Duin, 1998]).

Building on this work, Tran *et al*. [2014] proposed PSO-LSRG, an enhanced PSO variant incorporating two key improvements. First, a gbest restart mechanism activates when optimization stagnates, and second, a local search strategy applied to each particle's personal best solution maintains feature diversity. These modifications help prevent the premature discarding of potentially useful features.

In the domain of Cyber-Physical Systems IDSs, Quincozes *et al*. [2021] contributed the GRASP (Greedy Randomized Adaptive Search Procedure) algorithm. This approach combines constructive phases with local search to preserve relevant features. While we acknowledge this work's significance, we observe it doesn't address the increasingly important aspect of explainability in feature selection.

Our analysis of Vijayanand and Devaraj [2020]'s work reveals another approach to mitigating premature convergence in the Whale Optimization Algorithm (WOA). By integrating GA's crossover and mutation operators, the authors enhanced solution diversity. Their methodology initializes whale positions using randomly selected features, evaluates performance with an SVM classifier, and applies GA operators to generate new solutions. Results demonstrate improved attack detection rates compared to standard WOA and other evolutionary methods. However, we note the absence of XAI techniques like SHAP-based ranking that could provide more interpretable feature selection.

We identify key differences between these approaches and our work. While Vijayanand and Devaraj [2020] focuses on generational solution diversity, our method optimizes sequential feature exploration. This approach reduces redundancies and improves computational efficiency when selecting optimal feature subsets.

4 The IWSHAP-X Algorithm

We present IWSHAP-X (IWSHAP with eXploration), an enhanced algorithm that builds upon IWSHAP to overcome premature convergence limitations. The key innovation of our approach involves incorporating an additional exploration phase during feature selection. This extension improves intrusion detection performance by dynamically reassessing discarded features, thereby optimizing the search process and increasing the likelihood of discovering more effective feature combinations. Figure 1 provides a visual representation of the IWSHAP-X algorithm.

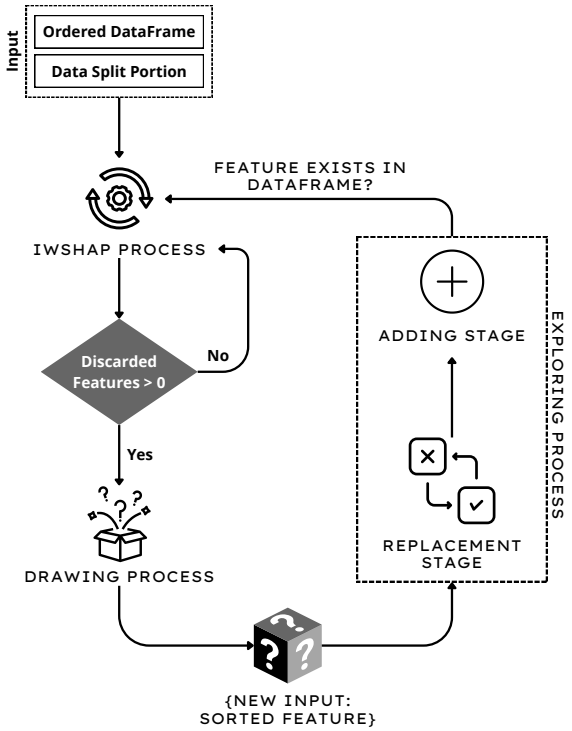


Figure 1. Illustration of the core of the IWSHAP-X algorithm

Our algorithm begins by establishing an initial feature rank-

ing based on SHAP values, which quantify each feature's relative importance. For this first step, we construct an ML model using the original dataset and compute SHAP values through the *Tree Explainer* method [ORG, 2024]. This prioritization ensures that more relevant features receive earlier consideration. The ranked dataset is then partitioned into training and testing subsets for subsequent processing.

The standard IWSHAP procedure continues iteratively until a feature discard event occurs. At this point, the *Drawing Process* (detailed in Section 4.1) activates, randomly selecting a new feature K according to SHAP-derived importance weights. This selected feature enters a replacement mechanism and rejoins the feature set during what we term the *Exploring Process*, which we describe fully in Section 4.2.

Beyond the SHAP-based weighting system, we integrate a non-repetition mechanism inspired by Tabu Search [Glover, Fred and Laguna, Manuel, 1998] to prevent redundant evaluation of previously examined feature subsets. This design choice minimizes unnecessary computations while encouraging the exploration of new feature combinations.

Regarding performance characteristics, we design IWSHAP-X to enhance model generalization capability while mitigating overfitting risks, ultimately yielding more accurate and robust feature selection explanations. Our choice of F1-Score as the evaluation metric ensures balanced consideration of both precision and recall - particularly valuable in class-imbalanced scenarios. The algorithm's sequential, incremental exploration of the solution space provides effective escape from local minima. Furthermore, as a wrapper-based approach, IWSHAP-X maintains compatibility with various ML models, representing a key advantage over more limited alternatives.

4.1 Drawing Process

We implement the Drawing Process (Algorithm 1) as a weighted random selection mechanism that uses SHAP values to guide feature exploration. This process ensures features are not selected purely at random, but rather according to their relative importance while still maintaining some probability of selecting less relevant features.

The process begins with the necessary preprocessing of SHAP values, as some features may lack proper numerical representations. We address this by replacing any missing values (`NaN`) with 0 (lines 3–4). This replacement is justified because, in the context of SHAP, a `NaN` indicates the absence of contribution from that feature rather than a numerical weight. Assigning a value of 0 ensures consistency in the Drawing Process, where numerical inputs are required for weight assignment while also preserving the semantic meaning that the feature has no influence on the model's decision. Alternative strategies, such as imputing arbitrary constants or discarding features, are intentionally avoided, as they could either introduce artificial bias or prematurely remove features that might become relevant in later exploration stages. By assigning 0, we preserve both mathematical validity and algorithmic fairness, ensuring that all features remain eligible for re-evaluation during subsequent analysis phases.

We then perform the weighting procedure (lines 5-10) through three key steps:

1. Assigning exponential weights based on feature importance rank;
2. Applying a decay rate controlled by the parameter α (set to 0.2 in lines 5-8);
3. Normalizing these weights creates a valid probability distribution (lines 9-10).

For the actual feature selection (lines 11-22), we first identify available features that are not currently in the best subset (lines 11-12). If none exist, the process terminates (lines 13-14). When features are available, we:

1. Compute their corresponding weights (lines 16-17);
2. Normalize these weights (lines 18-19);
3. Select the next feature probabilistically using these adjusted weights (line 20).

As a fallback mechanism, when no valid weights exist, we default to a uniform random selection among the available features (lines 21-22). This balanced approach provides intelligent feature selection that prioritizes important features while maintaining exploratory capability.

4.2 Exploring Process

In Algorithm 2, we detail the *Exploring Process* of IWSHAP-X. We first compute the best F1-Score ($f1_{\text{best}}$) from the current feature subset M and initialize the set of discarded features (those not in M) (lines 3-5). The process begins with the *Drawing* step (Section 4.1), where we randomly select a discarded feature K (lines 6-8).

Next, we proceed to the *Replacement* stage (lines 9-17). Here, the randomly chosen feature K replaces each feature in M sequentially. For each replacement, we retrain the model and evaluate the new subset using its F1-Score. If the new F1-Score exceeds $f1_{\text{best}}$, we update the current feature subset F and the best F1-Score accordingly (lines 13-16).

We then move to the *Adding* stage (lines 18-23), where we directly include K into the current feature subset. After retraining and evaluating the model with this expanded subset, we check whether the F1-Score improves. If it does, we update both the feature set and the best F1-Score.

Throughout the *Replacement* and *Adding* stages, the model undergoes repeated training and evaluation cycles. This allows us to systematically track performance metrics and retain the best-performing feature subset whenever an improvement occurs.

5 Experiments

To ensure a rigorous and accurate comparison, the experimental settings adopted in this study were aligned with those specified in the work that proposed the IWSHAP method [Scherer *et al*., 2024]. Below we present the materials and methods, followed by the results found.

5.1 Materials and Methods

The experiments were conducted on a machine equipped with an AMD Ryzen 7 5800X processor (8 cores), 64 GB of RAM,

Algorithm 1 Drawing Process

Input: DataFrame feature_importance_df, List feature_list, List best_features
Output: Selected feature next_feature
Replace NaN values with 0 in feature_importance_df[“importance”]
feature_importance_df[“importance”] ← feature_importance_df[“importance”].fillna(0)
Define exponential weights for sorted features
alpha ← 0.2 ▷ Decay rate for weights
indices ← np.arange(len(feature_importance_df))
shap_weights ← np.exp(−alpha × indices) ▷ Exponential decay
Normalize weights to sum to 1
shap_weights ← shap_weights/shap_weights.sum()
Select a feature based on adjusted weights
available_features ← [f for $f \in$ feature_list if $f \notin$ best_features]
if available_features $= \emptyset$ **then**
    break ▷ Stop if no available features
end if
available_weights ← [w for $f, w \in$ zip(feature_list, shap_weights) if $f \notin$ best_features]
if available_weights $\neq \emptyset$ **then**
    available_weights ← np.array(available_weights)
    available_weights ← available_weights/available_weights.sum() ▷ Normalize available weights
    next_feature ← random.choices(available_features, weights = available_weights, $k = 1$)[0]
else
    next_feature ← random.choice(available_features)
end if

Algorithm 2 IWSHAP-X Exploring Process

Input: Dataset D, Feature set F, Subset $M \subset F$, Model Mdl
Output: Updated feature set F', Best F1-Score $f1_{\text{best}}$
Initialize $f1_{\text{best}} \leftarrow$ compute_f1(Mdl, D, F)
$F' \leftarrow F$ ▷ Start with the original feature set
discarded_features $\leftarrow F \setminus M$ ▷ Track features not in subset M
while discarded_features $\neq \emptyset$ **do**
    $K \leftarrow$ randomly_select_feature(discarded_features) ▷ Randomly select a feature K from discarded features
    discarded_features $\leftarrow$ discarded_features $\setminus \{K\}$ ▷ Remove K from discarded features
    for each feature $m_i \in M$ **do** ▷ Iterate over the subset M
        $F_{\text{temp}} \leftarrow F'$ ▷ Create a temporary feature set
        Replace m_i with K in F_{temp} ▷ Substitute m_i with K
        $f1_{\text{temp}} \leftarrow$ compute_f1$(Mdl, D, F_{\text{temp}})$ ▷ Train and evaluate model
        if $f1_{\text{temp}} > f1_{\text{best}}$ **then**
            $F' \leftarrow F_{\text{temp}}$ ▷ Update feature set if F1-Score improves
            $f1_{\text{best}} \leftarrow f1_{\text{temp}}$ ▷ Update best F1-Score
        end if
    end for
    $F_{\text{temp}} \leftarrow F' \cup \{K\}$ ▷ Add K to the feature set
    $f1_{\text{temp}} \leftarrow$ compute_f1$(Mdl, D, F_{\text{temp}})$ ▷ Train and evaluate model
    if $f1_{\text{temp}} > f1_{\text{best}}$ **then**
        $F' \leftarrow F_{\text{temp}}$ ▷ Update feature set if F1-Score improves
        $f1_{\text{best}} \leftarrow f1_{\text{temp}}$ ▷ Update best F1-Score
    end if
end while

and running Ubuntu 22.04. To ensure the reproducibility of results, identical parameters were applied across all runs, and a consistent Docker environment configuration was maintained. The random seed (*random_state*) was fixed at 42, and the dataset was split into 80% for training and 20% for testing. After obtaining the selected feature subset with this split, a $5 - fold$ cross-validation procedure was employed to further evaluate the robustness and generalization capability of the selected features.

The experiments used the X-CANIDS [Jeong *et al*., 2024a] dataset, the same one employed in the original IWSHAP study Scherer *et al*. [2024]. This dataset is a publicly avail-

able collection of in-vehicle CAN-bus traffic specifically designed for intrusion detection research. It was captured from real vehicular environments and comprises 688 features extracted from CAN frames, including counters, checksums, ECU (Electronic Control Unit) status indicators, and sensor-related signals.

While the previous study focused on messages from a single segment, the present work investigates three CAN segments selected according to the lowest F1-scores reported by Jeong *et al*. [2024a]. These segments include both *Suspension* and *Masquerade* attacks, representing distinct intrusion behaviors in the network:

- *Suspension* ($S1$ - AID 2B0h): Targets messages related to the Steering Angle Sensor (SAS). By suspending these messages, the attacker disrupts the availability of steering angle data, which is critical for several safety mechanisms. This segment corresponds to the same scenario evaluated in the original IWSHAP paper.
- *Suspension* ($S2$ - AID 557h): nvolves additional suspension of diagnostic messages. This segment was included to increase sample diversity and to ensure a more representative evaluation. Suspension attacks primarily aim to disrupt legitimate traffic, leading to denial-of-service conditions in the CAN network.
- *Masquerade* (M - AID 381h): Injects spoofed messages that mimic legitimate ones from the Motor Driven Power Steering system. By falsifying steering-related signals, the attacker misleads ECUs into making unsafe decisions. Unlike suspension attacks, masquerade attacks compromise data integrity and authenticity rather than availability.

Analyzing both $S1$ and $S2$ enables a broader and more representative evaluation while preserving comparability with prior results. Moreover, analyzing M expands our scope to a different attack strategy. Table 2 presents the proportion of samples for each segment of the dataset used.

Table 2. Summary of Samples Used in the Study.

Attack (segment)	AID	Total Samples	Clean Samples	Attack Samples
Suspension (S1)	2B0h	784,774	688,780	95,994
Suspension (S2)	557h	784,774	775,173	9,601
Masquerade (M)	381h	784,774	736,776	47,998

Additionally, to validate the results and strengthen confidence in the robustness of the proposed algorithm, we applied a stratified $k - fold$ cross-validation procedure with $k = 5$. The experimental pipeline employed ordinal encoding for categorical variables while preserving numerical variables in their original form. Out-of-fold (OOF) predicted probabilities were used to generate Receiver Operating Characteristic (ROC) curves and to compute the Area Under the Curve (AUC) as the primary performance metric. Furthermore, confusion matrices were constructed at a decision threshold of 0.5 to analyze the trade-offs between false positives and false negatives. The results obtained under this evaluation protocol are presented and discussed in Section 5.4.

5.2 Detection and Explainability

In this section, we present the experimental results comparing the proposed IWSHAP-X method with the original IWSHAP algorithm. The evaluation focuses on three key aspects: classification performance (measured by the F1-Score), computational efficiency (execution time), and feature subset size. The experiments were conducted on two attack scenarios from the X-CANIDS dataset: *Suspension* (Subsection 5.2.1) and *Masquerade* (Subsection 5.2.2).It is important to note that in Tables 3 and 4, the Execution Time corresponds to the best-performing round. Specifically, among all the rounds executed by each algorithm, the one achieving the highest F1-Score was selected, and its execution time and corresponding feature set were reported as the final result for that algorithm.

5.2.1 Suspension Attacks

Table 3 summarizes the results for the *Suspension* scenarios ($S1$ and $S2$). As observed, IWSHAP-X outperforms IWSHAP across classification F1-Score, execution time, and feature subset size$S2$.

In the *Suspension ($S1$)* scenario, IWSHAP-X achieved an F1-Score of 91.92%, a marginal improvement over IWSHAP's 91.86%. However, the most significant gain lies in computational efficiency, with $\approx 58.5\%$ reduction in execution time (from 1.01s to 0.4193s). Moreover, IWSHAP-X selected a more compact feature subset, reducing the number of selected features from 19 to 11 while maintaining classification performance. Additionally, the best round for IWSHAP occurred at round 52 out of 688, where it achieved the highest F1-Score among all iterations and selected 18 features. This indicates that from round 52 onward, no subsequent combination of features yielded a higher F1-Score. For IWSHAP-X, the best round was 316, with 11 selected features, demonstrating its continued capacity for improvement throughout the iterative process.

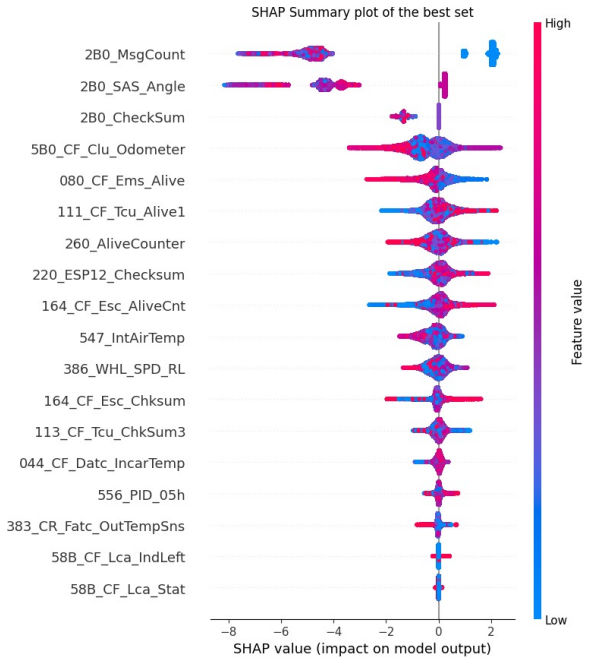
These results reinforce the impact of premature convergence in greedy-based feature selection methods. IWSHAP, by following an incremental selection process without an adequate exploration phase, quickly stabilizes at a suboptimal feature subset, limiting the discovery of potentially better feature combinations. In contrast, IWSHAP-X, equipped with an exploratory mechanism, effectively escapes this issue, finding smaller yet more efficient feature subsets.

To understand which features most influenced the classifier's decisions, we employed the SHAP library as an explainable AI technique. Figure 2a presents a SHAP summary plot generated from the feature subset selected by the IWSHAP method for the *Suspension ($S1$)* attack scenario. This plot highlights both the importance and the effect direction of each feature on the model's output.

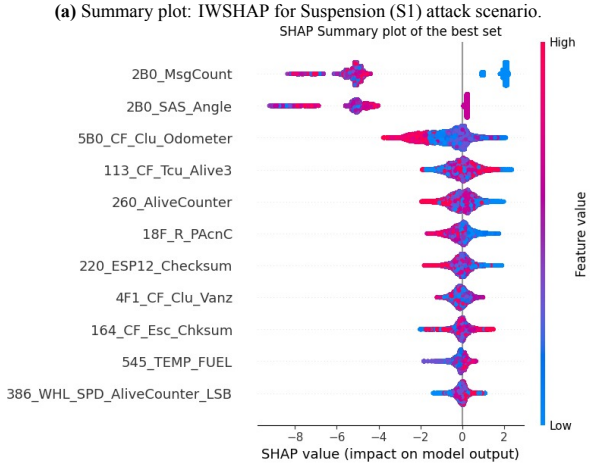
Features like `2B0_MsgCount`, `2B0_SAS_Angle`, and `5B0_CF_Clu_Odometer` show wide SHAP value distributions, demonstrating their strong impact on classification. The color gradient (blue for low values to red for high values) reveals how feature ranges affect predictions. For instance, higher values of `2B0_MsgCount` and `220_ESP12_Checksum` (red) correlate with attack predictions, while lower values (blue) suggest normal behavior. In contrast, features such

Table 3. Comparison of Results for Suspension Attack Scenarios.

Method	F1-Score		Execution Time (s)		# Features	
	Suspension (S1)	Suspension (S2)	Suspension (S1)	Suspension (S2)	Suspension (S1)	Suspension (S2)
IWSHAP-X	**91.92**	**90.63**	**0.4193**	**0.4668**	**11**	**14**
IWSHAP	91.86	89.44	1.0100	0.5283	19	21



(a) Summary plot: IWSHAP for Suspension (S1) attack scenario.



(b) Summary plot: IWSHAP-X for Suspension (S1) attack scenario.

Figure 2. Comparison of IWSHAP-X summary plots for Suspension attack (S1 scenario).

as `58B_CF_Lca_Stat` show minimal influence, indicating limited relevance for intrusion detection in this context.

Figure 2b compares the feature sets selected by both algorithms. While there's substantial overlap, IWSHAP-X produces a more compact subset (we analyze feature reduction in Section 5.3). Notably, some features exhibit opposite relationships between methods. For example, high values of `220_ESP12_Checksum` correlate with attacks in IWSHAP but show the inverse pattern in IWSHAP-X.

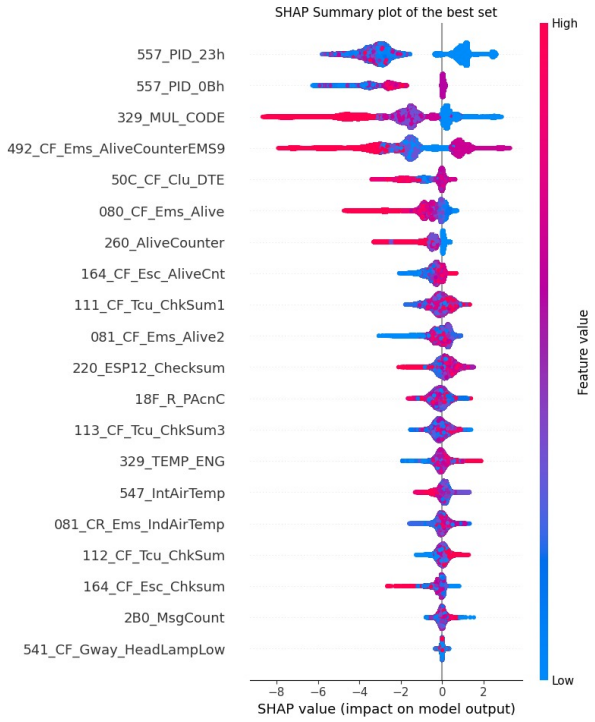
These differences emerge because SHAP values depend on feature context. When correlated features change, the model's interpretation of inputs can shift substantially. A feature might serve as a primary attack indicator in one configuration but play a secondary role in another. This effect is particularly pronounced in non-linear models like XGBoost, where decision paths adapt to the feature space.

For the *Suspension (S2)* scenario, IWSHAP-X achieves slightly better performance (90.63% F1-Score vs. 89.44%) while reducing the feature set by 33.3% (from 21 to 14 features). This leaner model shows improved resistance to overfitting. Execution time decreased by 11.7% (0.5283s to 0.4668s), though the reduction was less pronounced due to complex feature interactions in this subset. The corresponding summary plots appear in Figure 3a and Figure 3b.

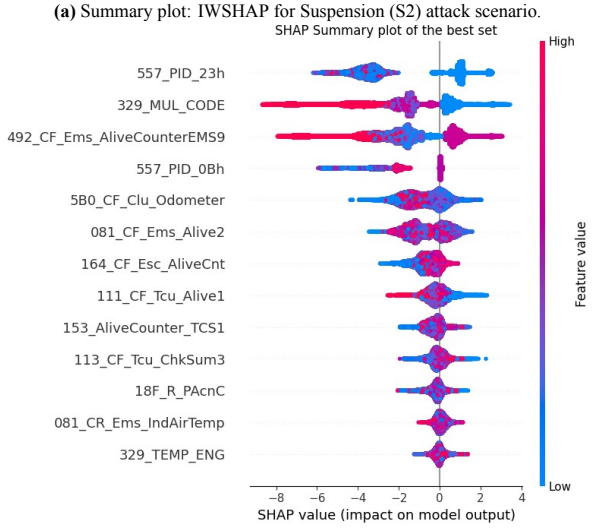
The analysis of the *Suspension* ($S2$) scenario reveals patterns consistent with our $S1$ observations. While both methods select overlapping feature subsets, we find notable variations in feature importance rankings between IWSHAP and IWSHAP-X. A clear example is `557_PID_0Bh`, which ranks as the second most important feature in IWSHAP but drops to fourth in IWSHAP-X. This difference shows how IWSHAP-X's enhanced exploration capability identifies more effective feature combinations, leading to revised contributions from individual features in the final classification.

Our results confirm that IWSHAP-X successfully overcomes the premature convergence limitation observed in IWSHAP. Both approaches achieve similar F1-Scores (90.63% vs 89.44%), but IWSHAP-X provides two significant advantages: (1) it reduces the feature set by 33.3% (from 21 down to 14 features), and (2) decreases execution time by 11.7%. These improvements make IWSHAP-X particularly valuable for real-time IDSs where computational efficiency and model simplicity are critical.

When comparing the two Suspension attack variants ($S1$ and $S2$), we observe that while some core features like `329_MUL_CODE`, `557_PID_23h`, and `492_CF_Ems_AliveCounterEMS9` remain important across both scenarios, their relative importance and SHAP value distributions vary significantly. This indicates that while these attacks share some common detection signals, each variant exhibits unique behavioral patterns in the CAN bus traffic. The model dynamically adjusts its feature weighting based on these differences, demonstrating the need for adaptive feature selection methods capable of capturing subtle variations between related attack types.



(a) Summary plot: IWSHAP for Suspension (S2) attack scenario.



(b) Summary plot: IWSHAP-X for Suspension (S2) attack scenario.

Figure 3. Comparison of SHAP summary plots using IWSHAP and IWSHAP-X for the Suspension (S2) attack scenario.

5.2.2 Masquerade Attack

Our evaluation of the Masquerade attack scenario (Table 4) reveals IWSHAP-X's superior feature selection capability. The method selects just 15 features compared to IWSHAP's 32 (53.13% reduction), while simultaneously improving classification performance from 94.43 to 98.55 F1-Score. This substantial enhancement comes with a 34.7% reduction in execution time (0.7349s to 0.4796s), demonstrating both improved accuracy and computational efficiency.

The Masquerade scenario highlights IWSHAP-X's particular effectiveness in complex detection tasks. Unlike the Suspension cases, we observe a more pronounced performance gap between methods, suggesting IWSHAP-X excels when dealing with numerous irrelevant features. Its dynamic exploration strategy better isolates meaningful patterns in the feature space, leading to more reliable predictions.

Table 4. Results for the Masquerade Attack Scenario.

Method	F1-Score	Execution Time (s)	Features
IWSHAP-X	**98.55**	**0.4796**	**15**
IWSHAP	94.43	0.7349	32

SHAP analysis, as illustrated in Figures 4 and 5, reveals several noteworthy patterns:

- Features such as `381_CF_Mdps_Stat` exhibit complex influence behaviors in both methods, where low values (blue) contribute to both attack and normal classifications;
- `111_CF_Tcu_Alive1` presents this dual contribution pattern exclusively in IWSHAP-X, indicating a more nuanced sensitivity to contextual variations;
- The feature importance rankings differ considerably between the two methods, showing greater divergence than in the Suspension scenarios.

These observations demonstrate IWSHAP-X's enhanced ability to capture subtle feature interactions, explaining its significant performance advantage (98.55 vs 94.43 F1-Score). The method's improved interpretability and detection capability make it particularly valuable for identifying Masquerade attacks, where subtle behavioral deviations are critical indicators.

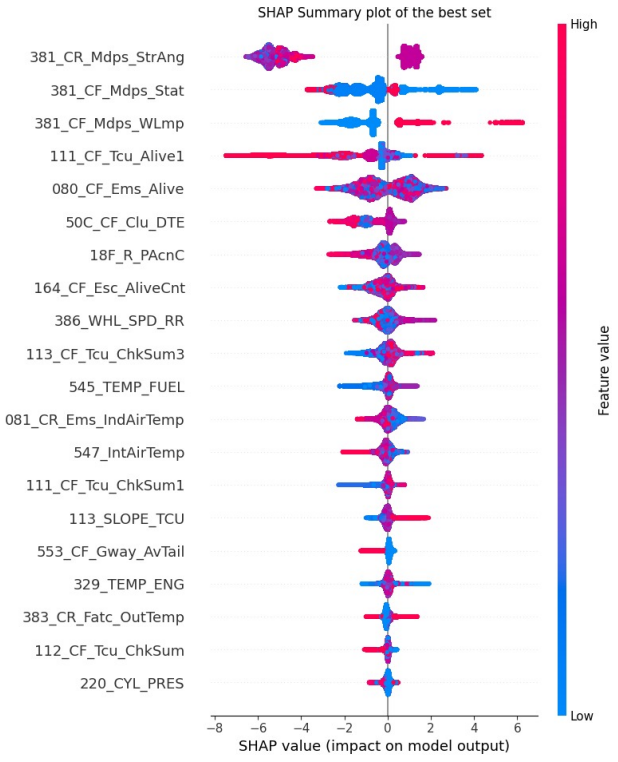


Figure 4. Summary plot: Masquerade attack with the IWSHAP method

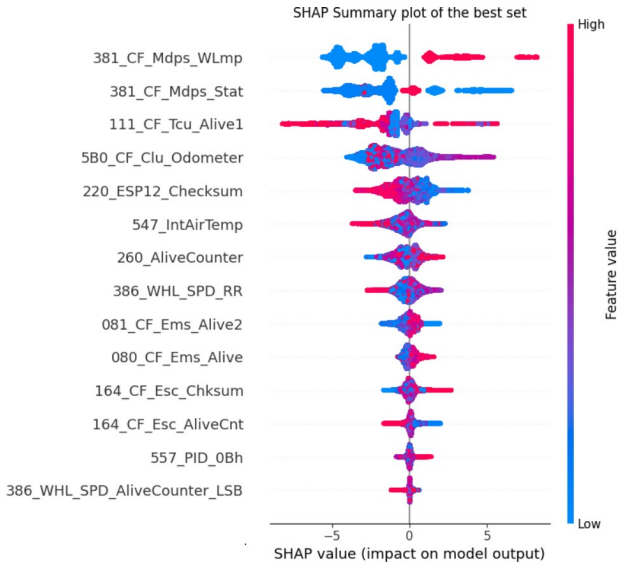


Figure 5. Summary plot: Masquerade attack with the IWSHAP-X method

5.2.3 Discussion

Our experimental results demonstrate IWSHAP-X's consistent advantages over the original IWSHAP method across all Suspension attack scenarios. While F1-Score improvements are incremental (e.g., 90.63% vs 89.44% for $S2$), the method achieves more substantial gains in computational efficiency ($\approx 58.5\%$ faster execution for $S1$) and feature reduction (33.3% fewer features for $S2$). These improvements validate the effectiveness of IWSHAP-X's enhanced exploration mechanism in avoiding premature convergence while maintaining detection accuracy.

The SHAP analysis provides deeper insights into these performance differences. We observe that:

- While both methods select overlapping feature sets, IWSHAP-X consistently identifies more optimal importance rankings;
- Some features exhibit reversed contribution patterns between methods (e.g., `220_ESP12_Checksum`);
- The reduced feature sets in IWSHAP-X show cleaner, more interpretable influence patterns.

These findings highlight IWSHAP-X's advantages: it not only produces more efficient models but also enhances interpretability by revealing clearer contribution patterns. This combination of performance and explainability makes IWSHAP-X particularly valuable for CAN IDSs, where both computational efficiency and interpretability are required.

5.3 Efficiency Analysis

The experimental results demonstrate IWSHAP-X's effectiveness in addressing premature convergence while enhancing feature selection efficiency. As shown in Figure 6, the algorithm consistently reduces the feature set size while maintaining or improving classification performance. Across the Suspension attack scenarios, we observe substantial feature reductions of 42.1% (from 19 to 11 features) for $S1$ and $\approx 33\%$ (from 21 to 14 features) for $S2$. These compact feature sets prove particularly valuable for CAN network IDSs operating under strict resource constraints.

The computational efficiency gains are equally noteworthy. Runtime reductions range from 11.7% in the Suspension $S2$ scenario to $\approx 58.5\%$ in the Suspension $S1$, with the Masquerade scenario achieving a $\approx 34.7\%$ improvement. These gains stem from IWSHAP-X's optimized exploration strategy, which minimizes redundant evaluations while converging more efficiently to optimal solutions.

It is important to note that the feature selection process, including repeated model retraining during exploration, is performed offline. Therefore, although this process introduces some computational overhead during development, it does not impact real-time deployment. The resulting models benefit from reduced feature sets and lower inference times, which ensure improved computational efficiency in production environments.

With respect to classification performance, IWSHAP-X occasionally achieves slightly higher F1-Scores compared to IWSHAP. Although these gains are encouraging, they should be interpreted as additional benefits rather than the primary goal of the method. The main contribution of IWSHAP-X lies in the reduction of the number of selected features and execution time, both of which represent consistent and classifier-independent improvements. In contrast, predictive performance depends heavily on the underlying learning algorithm (in this case, XGBoost), and therefore F1-Score gains cannot be guaranteed universally. Nevertheless, the observation that comparable or improved scores are achieved reinforces the notion that the efficiency improvements are obtained without compromising accuracy.

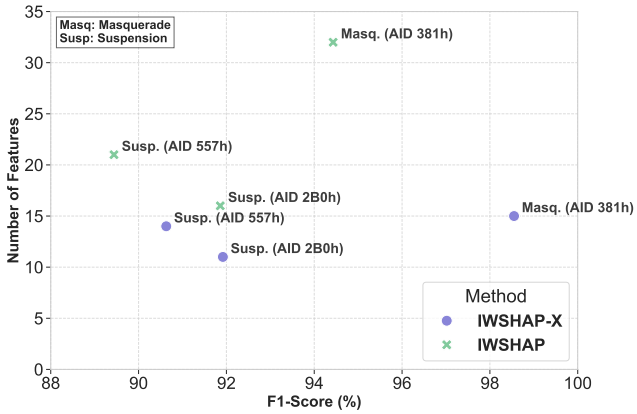


Figure 6. Scatterplot of method efficiency

Our explainability analysis reveals significant differences in feature interaction patterns between the methods. While both approaches select similar features, IWSHAP-X systematically modifies their importance rankings and sometimes even reverses their contribution directions in the final predictions. The SHAP summary plots clearly illustrate these differences - features like `220_ESP12_Checksum` in Suspension $S1$ and `557_PID_0Bh` in Suspension $S2$ show distinct influence patterns between the two methods.

The exploratory mechanism in IWSHAP-X provides dual benefits: it not only optimizes feature selection but also enhances model interpretability. By revealing more accurate feature interactions, it offers security analysts deeper insights into detection mechanisms. This transparency is crucial for building trust in security systems and developing effective

mitigation strategies.

These comprehensive results demonstrate IWSHAP-X's superiority over the original method across all evaluated metrics. The algorithm consistently identifies more compact feature sets while maintaining or improving detection accuracy, making it particularly valuable for real-time intrusion detection in resource-constrained environments like vehicular networks. Its balanced combination of efficiency, performance, and explainability represents a significant advancement for security applications.

5.4 Predictive Performance Analysis

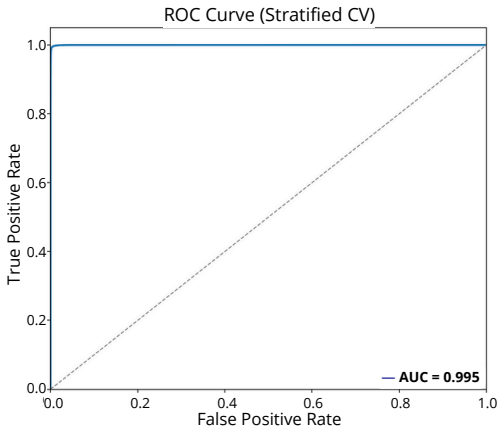
In the previous sections, we assessed validation primarily through the number of selected features, processing time, and single-run F1-Scores. We now turn to a stricter evaluation of predictive performance under the protocol described in Section 5.1: stratified k-fold cross-validation with $k = 5$, construction of out-of-fold (OOF) probabilities for ROC calculation, and confusion matrices computed at the fixed threshold of 0.5. Results are reported in Table 5, complemented by the confusion matrices in Figure 10 and ROC curves for each scenario. To further illustrate predictive stability across folds, we also provide Figure 12, which depicts per-fold F1 differences, paired fold-wise comparisons, and a boxplot for each dataset. The interpretation throughout accounts for the class priors summarized in Table 2: $S1$ is moderately imbalanced ($\approx 12.2\%$ attacks), $S2$ is severely imbalanced ($\approx 1.2\%$), and M has a low but non-negligible attack prevalence ($\approx 6.1\%$).

A first inspection reveals that IWSHAP-X either matches or outperforms IWSHAP across datasets. The improvements are minimal on $S1$, modest on $S2$, and most pronounced on M. Since AUC ROC values are already close to saturation (≥ 0.995) for both approaches, the most discriminative insights emerge from threshold-dependent metrics (Precision, Recall, F1), which are more informative than Accuracy in the imbalanced settings considered.

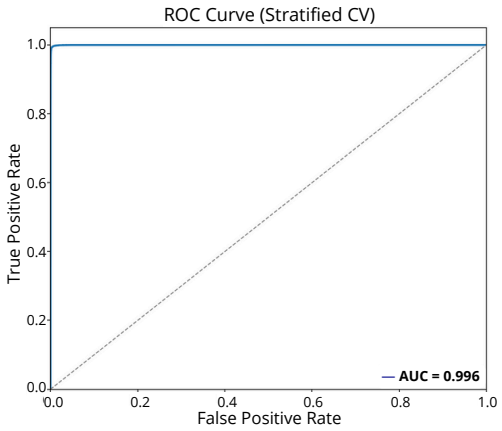
5.4.1 S1 Dataset

For $S1$, the two methods yield almost indistinguishable predictive behavior. According to Table 5, IWSHAP reaches an F1-Score of 0.9079, compared with 0.9072 for IWSHAP-X. Precision and Recall pairs remain well balanced and nearly identical ($0.9082/0.9076$ for IWSHAP vs. $0.9052/0.9092$ for IWSHAP-X), with Accuracy also practically the same (0.9775 vs. 0.9772). AUC ROC values are saturated at 0.996 (IWSHAP) and 0.995 (IWSHAP-X), with overlapping curves (Figures 7a, 7b), reflecting that both approaches induce equivalent instance rankings.

Taken together, these results show that in $S1$ the differences between IWSHAP and IWSHAP-X are within expected sampling variability under cross-validation. The small F1-gap ($\Delta \approx 0.0007$) is operationally negligible and unlikely to be statistically significant. Importantly, the balanced class prior in $S1$ ($\approx 12.2\%$ attacks) ensures that both algorithms operate in a regime where Precision and Recall trade-offs are stable across thresholds. Therefore, from a predictive standpoint, the methods can be considered tied. The practical advantage of IWSHAP-X lies in its reduced feature set and lower computational cost (Section 5.3), which it achieves without degrading accuracy.



(a) IWSHAP-X



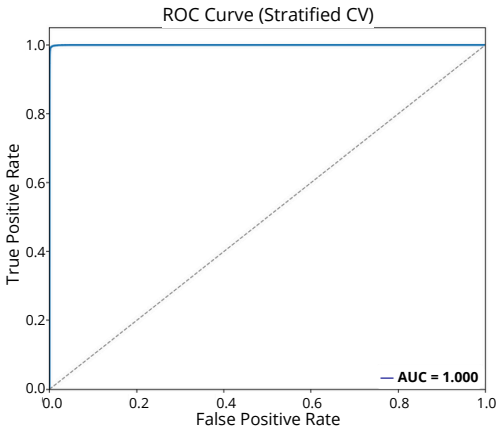
(b) IWSHAP

Figure 7. AUC ROC Curves - $S1$

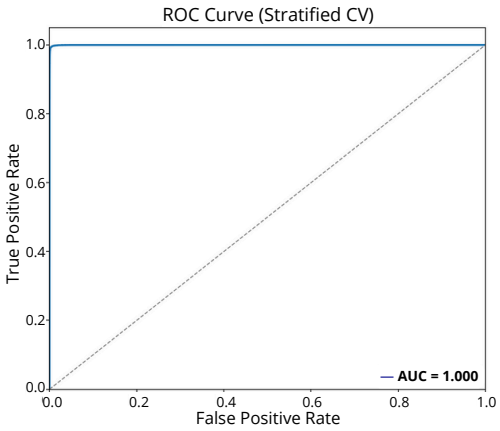
5.4.2 S2 Dataset

On $S2$, both algorithms behave as nearly perfect rankers, with AUC ROC saturating at 1.0 for both (Table 5). However, when applying the fixed threshold of 0.5, systematic differences emerge. IWSHAP-X achieves an F1-Score of 0.8975, while IWSHAP attains 0.8904. This improvement is driven primarily by a higher Precision from IWSHAP-X (0.8783 vs. 0.8656), while Recall remains similar (0.9176 vs. 0.9168). Accuracy is saturated in both settings (0.9974 for IWSHAP-X and 0.9972 for IWSHAP), confirming that it provides limited information under severe imbalance, given the low prior prevalence of attacks ($\approx 1.2\%$). Figure 8 shows both algorithm's ROC curves.

From an operational perspective, these results are meaningful. Because $S2$ is the most imbalanced of the three scenarios, every reduction in false positives has considerable impact: for in-vehicle IDS deployments, even modest gains in Precision translate into fewer false alarms per million frames. The consistent but small F1 advantage of IWSHAP-X thus reflects a calibrated improvement over IWSHAP, without compromising Recall. Taken together with the reductions in feature set size and execution time reported earlier (Sections 5.2 and 5.3), $S2$ highlights how the exploratory mechanism of IWSHAP-X converts equivalent ranking ability into a more favorable operating point under extreme imbalance.

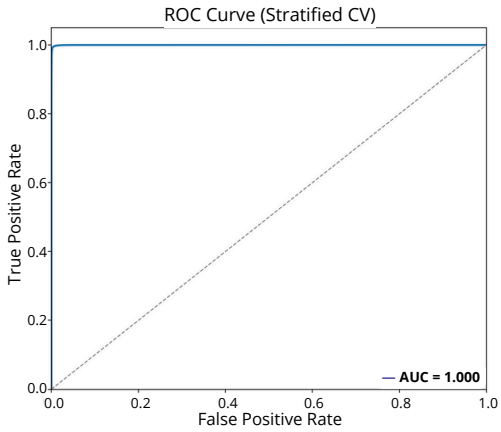


(a) IWSHAP-X

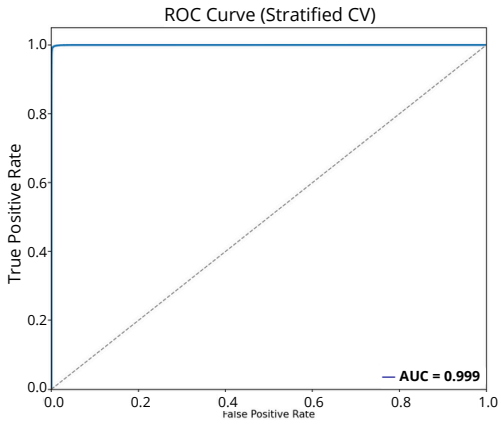


(b) IWSHAP

Figure 8. AUC ROC Curves - $S2$



(a) IWSHAP-X



(b) IWSHAP

Figure 9. AUC ROC Curves - M

5.4.3 M Dataset

The M scenario reveals the clearest separation between the two methods. At the aggregate level, IWSHAP-X attains near-perfect performance across all metrics: F1-Score $= 0.9834$, Precision $= 0.9818$, Recall $= 0.9850$, Accuracy $= 0.9980$, and AUC ROC $= 1.0000$ (Table 5). By contrast, IWSHAP achieves substantially lower thresholded scores, with F1-Score $= 0.9462$, Precision $= 0.9382$, Recall $= 0.9544$, Accuracy $= 0.9934$, and AUC ROC $= 0.9999$. The performance gap in F1-Score ($\Delta = 0.0372$) reflects the joint effect of improved Precision and Recall in IWSHAP-X, translating into a marked reduction in both false positives and false negatives when compared to IWSHAP (Figure 10).

These observations align with the explainability analysis in Section 5.2.2, where SHAP inspection revealed that IWSHAP tends to retain weaker or unstable predictors in this scenario. By contrast, IWSHAP-X places importance on features such as Alive counters, checksums, and steering-related signals that provide more robust class separation. From an IDS perspective, this difference is critical: while both methods can rank instances effectively (nearly perfect AUC scores), IWSHAP miscalibrates scores at the decision threshold, leading to an excess of false alarms. IWSHAP-X avoids this failure mode, producing both extremely high Recall and substantially higher Precision, a combination that directly translates into fewer unnecessary interventions in deployment while maintaining coverage of true attacks.

Overall, these results reinforce the central claim of this work: while both methods produce strong instance rankings, the exploratory mechanism in IWSHAP-X yields models that are more parsimonious, computationally efficient, and better aligned with operational needs. In balanced scenarios like $S1$, performance is equivalent but efficiency improves; in skewed scenarios like $S2$ and especially M, IWSHAP-X translates strong ranking capacity into tangible thresholded performance improvements. This combination of robustness, efficiency, and practical relevance highlights IWSHAP-X as a consistent advancement for CAN-bus intrusion detection.

5.4.4 Statistical Validation and Fold-wise Results

To verify that the observed improvements are not attributable to sampling variation, we performed paired significance testing on the cross-validated F1-Scores. Let $d_i = \mathrm{F1}_{\text{IWSHAP-X},i} - \mathrm{F1}_{\text{IWSHAP},i}$ for folds $i = 1, \ldots, 5$. Figure 12 summarizes the per-fold differences, paired comparisons, and F1-Score distributions across all three datasets.

The inferential analyzes provide a more detailed understanding of the fold-wise results for each dataset. Figure 11 presents a condensed overview of the key findings.

For $S1$, the mean F1-Scores are nearly identical (0.9079 ± 0.0036 for IWSHAP and 0.9072 ± 0.0037 for IWSHAP-X), with an average fold-wise difference of -0.0007. The paired t-test indicates no systematic advantage ($t(4) = -0.2212$, $p = 0.8358$), and the Wilcoxon signed-rank test similarly reveals no detectable effect ($W = 7.0000$, $p = 1.0000$). These results are consistent with the per-fold visualizations,

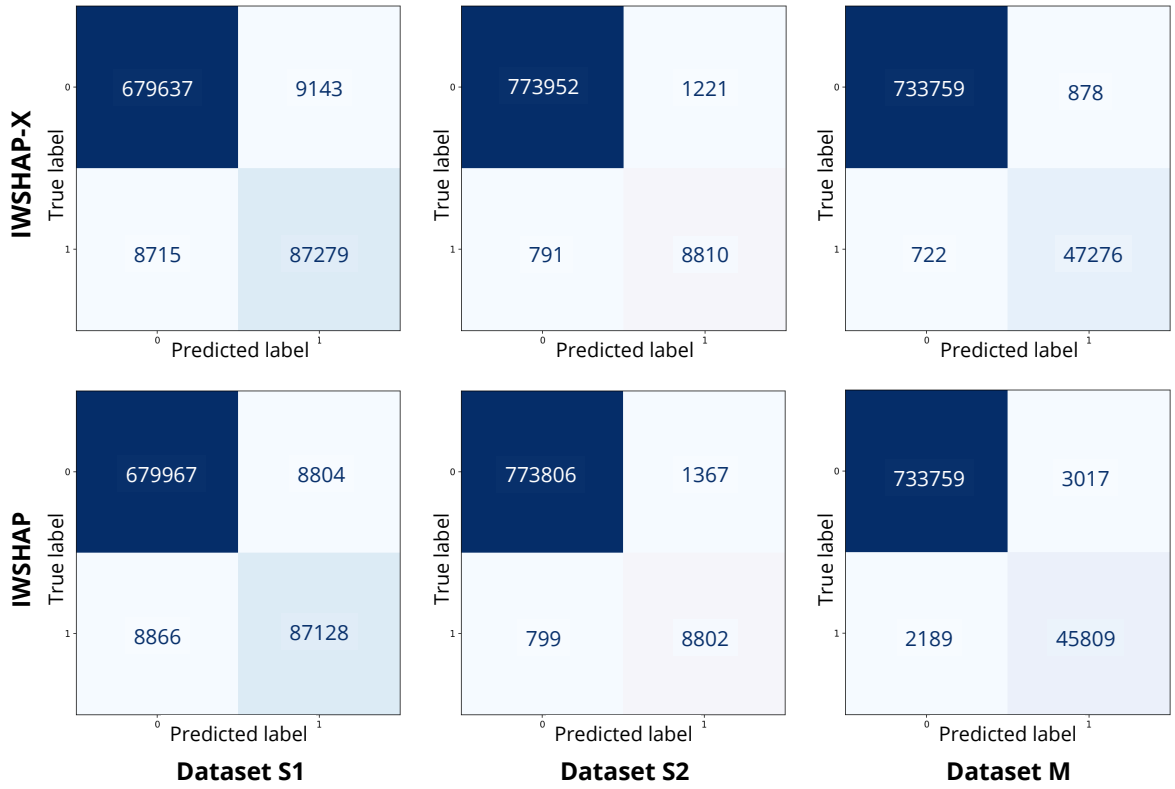


Figure 10. Confusion Matrices

Algorithm	Attack	F1-Score	Precision	Recall	Accuracy	AUC ROC
IWSHAP-X	S1	0.9072	0.9052	0.9092	0.9772	0.9950
	S2	0.8975	0.8783	0.9176	0.9974	1.0000
	M	0.9834	0.9818	0.9850	0.9980	1.0000
IWSHAP	S1	0.9079	0.9082	0.9076	0.9775	0.9960
	S2	0.8904	0.8656	0.9168	0.9972	1.0000
	M	0.9462	0.9382	0.9544	0.9934	0.9999

Table 5. Summary of Metrics After K-Fold

which show alternating dominance between methods without a stable trend.

For $S2$, IWSHAP-X demonstrates a modest mean improvement (0.8975 ± 0.0092) over IWSHAP (0.8904 ± 0.0025), resulting in an average difference of $+0.0071$. The paired t-test suggests a trend favoring IWSHAP-X, though not statistically significant under conventional thresholds ($t(4) = 1.8268$, $p = 0.1418$). The Wilcoxon signed-rank test supports this interpretation ($W = 2.0000$, $p = 0.1875$). Combined with the fold-wise plots, these results indicate that IWSHAP-X attains a slightly more favorable operating point under class imbalance, although some variability remains across folds.

The case of M stands in clear contrast. IWSHAP-X attains a substantially higher mean F1-Score (0.9834 ± 0.0006) compared to IWSHAP (0.9462 ± 0.0011), for an average improvement of $+0.0371$. The paired t-test confirms a highly significant effect ($t(4) = 55.4514$, $p < 0.0001$), with improvements consistent across all folds. Although the Wilcoxon signed-rank test again reports $W = 0.0000$, $p = 0.0625$ — reflecting discreteness under small samples ($n = 5$) — the unanimity of fold-wise gains and the extremely large t-statistic provide strong evidence of systematic superiority.

In summary, statistical tests reinforce the narrative from per-fold plots: $S1$ shows equivalence, $S2$ exhibits modest but consistent improvements in favor of IWSHAP-X, and M demonstrates a decisive and highly significant advantage. These findings, when combined with efficiency gains and feature reductions (Sections 5.2 and 5.3), confirm that IWSHAP-X delivers not only computational benefits but also tangible predictive improvements in the most challenging and operationally relevant scenarios.

The fold-level visualizations provide further insight:

- For $S1$ (first row, Figure 12), the per-fold differences fluctuate around zero. Some splits favor IWSHAP, others IWSHAP-X, with paired lines crossing frequently and boxplots showing nearly identical medians. This confirms that on $S1$ the two methods are effectively tied, with differences attributable to threshold effects rather than systematic ranking strength.
- For $S2$ (second row), the plots indicate a more consistent improvement for IWSHAP.X. Four out of five folds show higher F1-Scores, with gains of up to ≈ 0.016, while only one fold slightly favors IWSHAP. The paired lines rise in most cases, and the boxplot shows a higher

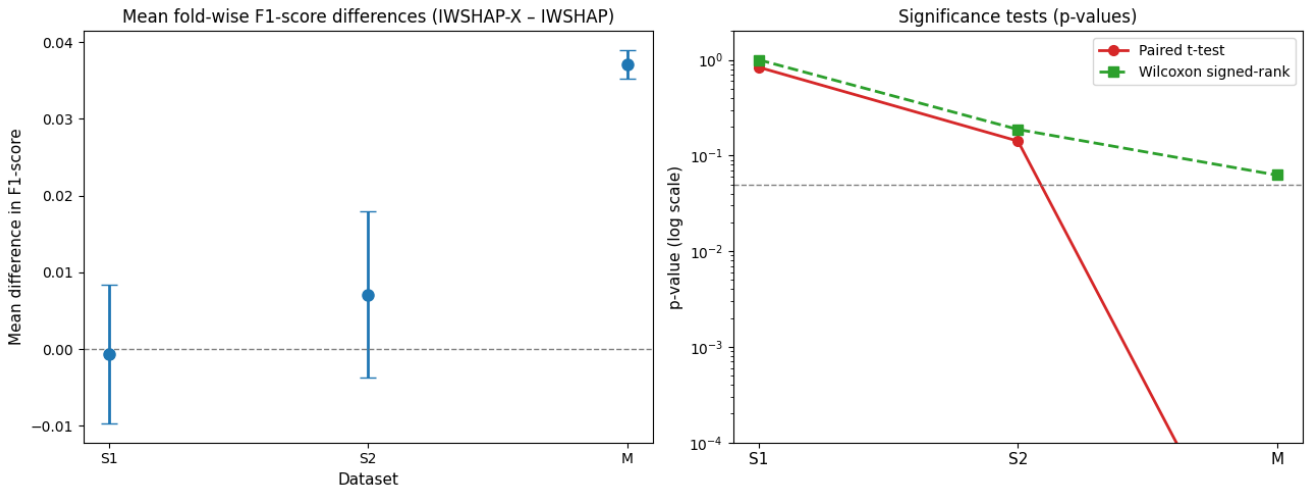


Figure 11. Statistical Significance Tests: t-test and Wilcoxon

median and reduced dispersion for IWSHAP-X, suggesting a modest performance uplift and greater fold-to-fold stability under severe class imbalance.

- For M (third row), the superiority of IWSHAP-X is evident. The per-fold differences are consistently positive, ranging from ≈ 0.034 to 0.040. The paired comparison lines exhibit a clear upward slope, and the boxplots reveal both a higher median and a substantially tighter distribution for IWSHAP-X. These results indicate not only improved average performance but also greater stability and reliability across folds, in contrast to the weaker and more variable outcomes of IWSHAP.

We include this analysis as a dedicated subsection rather than splitting it across $S1$, $S2$, and M because the hypothesis tests operate on fold-level paired observations and summarize the systematic behavior of the methods under the common cross-validation protocol. This placement avoids repetition, keeps the per-scenario narratives in Sections 5.4.1 - 5.4.3 focused on domain-specific error profiles, and concentrates the statistical evidence where it is most interpretable.

Takeaways. Across datasets, AUC ROC values remain close to saturation (≥ 0.995), indicating that both methods achieve highly separable class rankings. The main distinctions become evident only after applying the threshold of 0.5. The scenario-specific analyses, combined with the formal statistical validation, reveal a consistent set of insights:

- *On $S1$:* IWSHAP and IWSHAP-X perform essentially equivalently. The aggregate F1-Scores differ only at the third decimal place (0.9079 vs. 0.9072), the fold-wise plots show alternating dominance with no systematic advantage, and the statistical tests confirm equivalence ($t(4) = -0.2212$, $p = 0.8358$; Wilcoxon $W = 7.0000$, $p = 1.0000$). In this setting, the advantage of IWSHAP-X lies not in classification accuracy but in its reduced feature set and lower execution cost.
- *On $S2$:* IWSHAP-X provides modest yet systematic improvements. Its higher Precision (0.8783 vs. 0.8656) reduces false positives under severe class imbalance, with four out of five folds favoring IWSHAP-X and improvements of up to ≈ 0.016. Nevertheless, statistical testing reflects the limited sample size and variability across folds: the paired t-test does not reach conventional significance ($t(4) = 1.8268$, $p = 0.1418$), and the Wilcoxon test leads to the same conclusion ($W = 2.0000$, $p = 0.1875$). From an operational perspective, however, even these small Precision gains are relevant, as they translate into fewer false alarms in high-volume CAN-bus traffic.
- *On M:* IWSHAP-X exhibits clear and statistically robust superiority. The mean F1-Score rises from 0.9462 ± 0.0011 (IWSHAP) to 0.9834 ± 0.0006, yielding a difference of $+0.0371$. Fold-wise plots show unanimous gains within the range of $0.034 - 0.040$, and the results are strongly supported by the paired t-test ($t(4) = 55.4514$, $p < 0.0001$). Although the Wilcoxon test again reports $p = 0.0625$ due to the discreteness associated with $n = 5$, it yields $W = 0.0000$ (all folds favoring IWSHAP-X), providing convergent evidence of systematic improvement.

Overall, while both methods deliver strong separability at the ranking level, the exploratory mechanism of IWSHAP-X ensures more robust calibration at the decision threshold. In balanced conditions such as $S1$, predictive accuracy is essentially equivalent, but efficiency improves. Under severe imbalance ($S2$), modest yet consistent Precision gains reduce operational false alarms. In more complex scenarios such as M, IWSHAP-X yields overwhelming improvements in both Precision and Recall while maintaining stability across folds. These findings confirm that IWSHAP-X not only mitigates premature convergence and reduces resource demands but also translates ranking strength into practical enhancements in thresholded performance, where it matters most for vehicular IDS deployment.

5.5 Interpretability of Attacks

Like its predecessor, IWSHAP-X maintains the ability to interpret attacks through SHAP plots while delivering more precise results due to its enhanced feature selection. This capability becomes particularly powerful when combined with

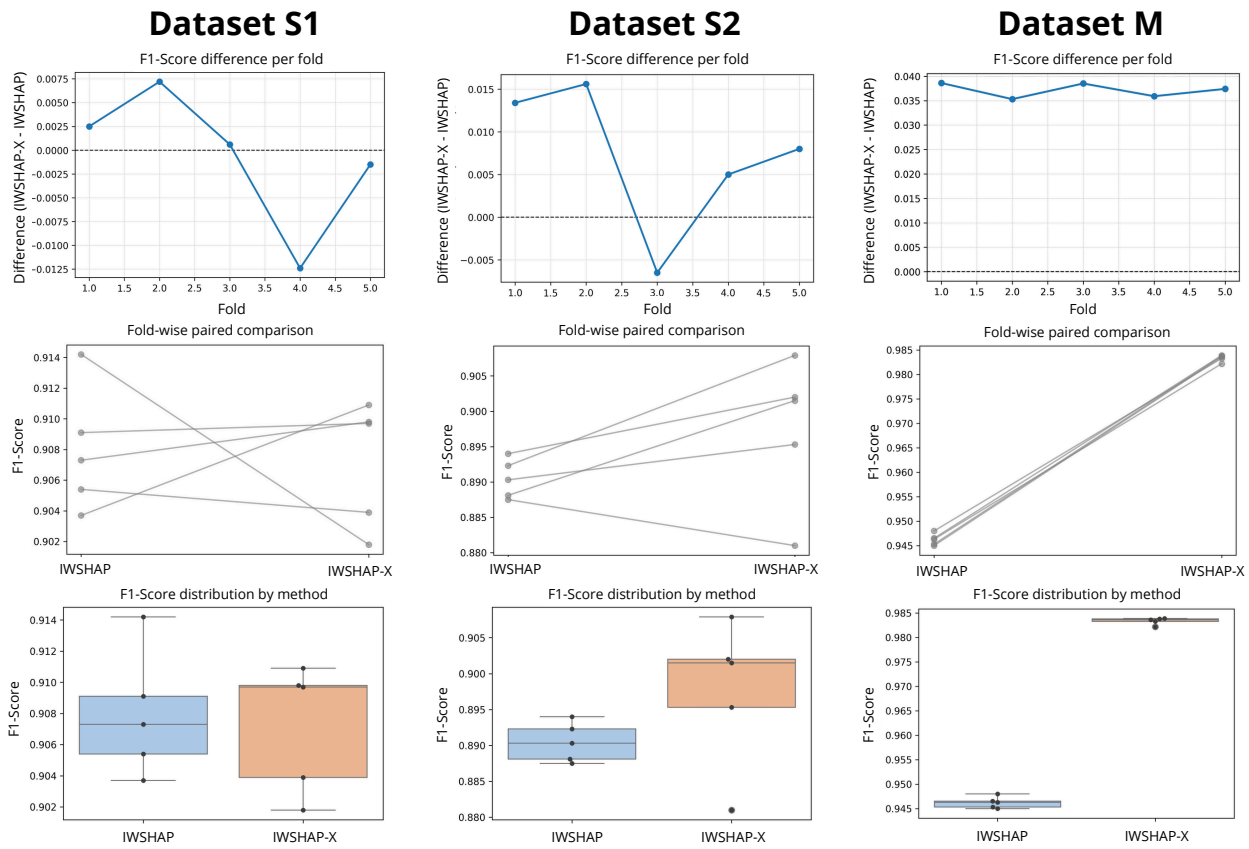


Figure 12. Fold-wise Results

domain knowledge about vehicle communication networks. Figure 13 shows some of the most affected ECUs identified.

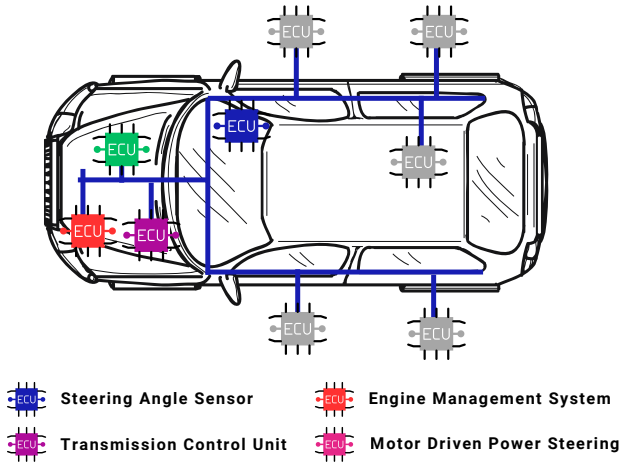


Figure 13. Most Affected ECUs

Recent work by Dresch *et al*. [2024] provides crucial context by analyzing the X-CANIDS dataset [Jeong *et al*., 2024a] to clarify feature meanings. Their approach enables mapping attacker behavior to specific vehicle functions. The feature naming convention itself reveals valuable information: the first component represents the Arbitration ID (AID), indicating message priority (lower values = higher priority), followed by the related ECU and signal purpose. For example, in `381_CF_Mdps_WLmp`, 381 is the AID.

IWSHAP-X's more reliable outputs significantly improve ECU compromise analysis. In the first suspension attack, both methods agree on the top two features (`2B0_Msg_Count` and `2B0h_SAS_Angle`), but differ on the third. IWSHAP selects `2B0h_CheckSum` while IWSHAP-X identifies `5B0_CF_Tcu_Alive3` as more important. The latter is likely more accurate given IWSHAP-X's improved selection mechanism. Notably, features containing 'Count' typically monitor message frequency, while 'Checksum' refers to data integrity verification - both useful for anomaly detection.

For the second suspension attack, both methods share two top features (`557_PID_23h` and `329_MUL_CODE`) but differ in their ordering and third selection. IWSHAP includes `557_PID_0Bh` while IWSHAP-X prioritizes `492_CF_Ems_AliveCounterEMS9`. PID features generally relate to diagnostic operations, while 'Alive' typically indicates ECU status monitoring. The 'MUL' designation appears manufacturer-specific and requires additional context for full interpretation.

The masquerade attack results show the most significant divergence. While both methods share two features in their top three (`381_CF_Mdps_Stat` and `111_CF_Tcu_Alive1`), they disagree on the most important feature. IWSHAP ranks `381_CRMdps_StrAng` (steering angle data) highest, while IWSHAP-X identifies `381_CF_Mdps_WLmp` (MDPS warning light status) as most critical. This difference highlights IWSHAP-X's ability to surface more operationally relevant features for attack detection.

These insights enable security analysts to not only understand which features influence detection most, but also assess their real-world implications. By distinguishing between fea-

tures representing critical vehicle functions versus less critical ones, we can better evaluate the actual risks posed by detected attacks and prioritize mitigation efforts accordingly.

6 Conclusion

Our work presents IWSHAP-X, an enhanced feature selection method that addresses the premature convergence limitation of IWSHAP through a novel metaheuristic exploration mechanism. By combining dynamic SHAP-based feature ranking with iterative refinement, the method identifies more compact and discriminative feature subsets while improving computational efficiency and model interpretability.

The experimental results on the X-CANIDS dataset demonstrate IWSHAP-X's consistent advantages. The algorithm achieves feature reductions of up to 53.13% while maintaining or improving detection performance across all evaluated attack scenarios. The corresponding computational efficiency gains make it particularly suitable for real-time intrusion detection in resource-constrained environments like vehicular networks.

These findings underscore the effectiveness of IWSHAP-X in the CAN environment. Several promising directions emerge for future research. Among them, we highlight extending the application of IWSHAP-X to broader cybersecurity domains beyond CAN systems, as well as evaluating its robustness against adversarial attacks targeting feature selection. In addition, we plan to conduct more comprehensive comparisons with additional state-of-the-art methods to strengthen the comparative analysis. Aligned with these initiatives, we also intend to investigate alternative metaheuristic approaches to further optimize the method's performance.

We recognize that although IWSHAP-X has a general structure and can be applied to diverse supervised learning scenarios, such as healthcare, finance, or industrial monitoring, this study intentionally focused on the CAN network environment. This choice was driven by the relevance of vehicular security and the demanding nature of the X-CANIDS dataset, which poses significant challenges for feature selection algorithms due to its scale and complexity. For future work, we plan to extend the evaluation to additional domains by incorporating datasets from distinct application areas, exploring a broader range of attack scenarios, and expanding the set of baseline algorithms..

We particularly emphasize the importance of incorporating domain knowledge into the feature selection process. This integration not only enhances detection performance but also improves model transparency, which is a critical requirement for modern security systems. As intrusion detection systems increasingly demand both high accuracy and interpretability, methods such as IWSHAP-X that effectively balance these objectives are expected to become essential tools for critical cybersecurity applications.

Declarations

Authors' Contributions

F.H.S., F.N.D., and M.M.C. contributed to the conceptualization, implementation, and execution of the experiments. S.E.Q. and D.K. provided crucial methodological insights and supervised the research process. V.E.Q. contributed to the development of the theoretical framework and the critical review of the manuscript. All authors actively participated in the discussion of the results, the drafting of the manuscript, and the final approval of the submitted version.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

We gratefully acknowledge the financial support provided by the Programa de Desenvolvimento Acadêmico (PDA) at Universidade Federal do Pampa (UNIPAMPA) and Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) through research scholarships and infrastructure funding. Our work benefited significantly from the computational resources made available by UNIPAMPA's Laboratory of Advanced Studies in Computing (LEA). We also extend our sincere appreciation to our colleagues and the anonymous reviewers whose valuable insights and constructive feedback helped strengthen this research.

Funding

This work was supported by the Programa de Desenvolvimento Acadêmico (PDA) at Universidade Federal do Pampa (UNIPAMPA) through research scholarships. Additional funding was provided by the Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) under grants 24/2551-0001368-7 and 24/2551-0000726-1, which supported infrastructure and research development. The authors gratefully acknowledge this financial support that made the research possible.

Availability of data and materials

The dataset analyzed during this study is publicly available from its source (X-CANIDS Jeong *et al*. [2024b], although access or redistribution may be subject to their respective terms of use. IWSHAP-X is publicly available in the following repository: `https://github.com/felipehscherer/iwshap-x`. For further information or access, please contact the corresponding author.

References

Abdulganiyu, O. H., Tchakoucht, T. A., and Saheed, Y. K. (2024). Towards an efficient model for network intrusion detection system (ids): systematic literature review. *Wireless Networks*, 30(1):453–482. DOI: 10.1007/s11276-023-03495-2.

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Master's thesis, Chalmers University of Technology. Available at:`https://www.cse.msu.edu/~cse960/Papers/security/axelsson00intrusion.pdf`.

Azam, Z., Islam, M. M., and Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision

tree. *IEEE Access*, 11:80348–80391. DOI: 10.1109/ACCESS.2023.3296444.

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). DOI: 10.3390/s23073610.

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. DOI: 10.1109/CIDM.2009.4938673.

Buiya, M. R., Laskar, A., Islam, M. R., Sawalmeh, S., Roy, M., Roy, R., and Sumsuzoha, M. (2024). Detecting iot cyberattacks: advanced machine learning models for enhanced security in network traffic. *Journal of Computer Science and Technology Studies*, 6(4):142–152. DOI: 10.32996/jcsts.2024.6.4.16.

Cai, J., Luo, J., Wang, S., and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79. DOI: 10.1016/j.neucom.2017.11.077.

Chaitanya, K., Somayajulu, D. V., and Krishna, P. R. (2021). Memory-based approaches for eliminating premature convergence in particle swarm optimization. *Applied Intelligence*, 51:4575–4608. DOI: 10.1007/s10489-020-02045-z.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & electrical engineering*, 40(1):16–28. DOI: 10.1016/j.compeleceng.2013.11.024.

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. DOI: 10.23919/MIPRO.2018.8400040.

Dresch, F., Scherer, F., Quincozes, S., and Kreutz, D. (2024). Modelos interpretáveis com inteligência artificial explicável (xai) na detecção de intrusões em redes intraveiculares controller area network (can). In *Anais do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 445–460, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbseg.2024.241421.

Duin, R. (1998). Multiple Features. DOI: 10.24432/C5HC70.

Esseghir, M. A. (2010). Effective wrapper-filter hybridization through grasp schemata. Avaialble at:`https://proceedings.mlr.press/v10/esseghir10a.html`.

Fatema, K., Dey, S. K., Anannya, M., Khan, R. T., Rashid, M. M., Su, C., and Mazumder, R. (2025). Federated xai ids: An explainable and safeguarding privacy approach to detect intrusion combining federated learning and shap. *Future Internet*, 17(6):234. DOI: 10.3390/fi17060234.

Glover, Fred and Laguna, Manuel (1998). *Tabu search*. Springer. DOI: 10.1007/978-1-4419-7997-$1_1$7.

Guvenir, H. and Quinlan, R. (1997). Arrhythmia. DOI: 10.24432/C5BS32.

Guyon, I. (2004). Madelon. DOI: 10.24432/C5602H.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182. DOI: 10.5555/944919.944968.

Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., Ahmad, I., and Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*, 110:102448. DOI: 10.1016/j.cose.2021.102448.

Jeong, S., Lee, S., Lee, H., and Kim, H. K. (2024a). 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. DOI: 10.1109/TVT.2023.3327275.

Jeong, S., Lee, S., Lee, H., and Kim, H. K. (2024b). 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. DOI: 10.1109/TVT.2023.3327275.

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. DOI: 10.1002/ett.4970.

Lamsaf, A., Carrilho, R., Neves, J. C., and Proença, H. (2025). Causality, machine learning, and feature selection: a survey. *Sensors*, 25(8):2373. DOI: 10.3390/s25082373.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2017). Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45. DOI: doi.org/10.1145/3136625.

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. DOI: 10.1186/s40537-024-00892-y.

Madhloom Kurdi, W. H., Alzuabidi, I. A., Najim, A. H., Kadhim, M. N., and Ahmed, A. A. (2025). Efficient two-stage intrusion detection system based on hybrid feature selection techniques and machine learning classifiers. *International Journal of Intelligent Engineering & Systems*, 18(3). DOI: 10.22266/ijies2025.0430.16.

Mallidi, S. K. R. and Ramisetty, R. R. (2025). Optimizing intrusion detection for iot: a systematic review of machine learning and deep learning approaches with feature selection and data balancing. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 15(2):e70008. DOI: 10.1002/widm.70008.

Mitchell, R. and Chen, I.-R. (2014). A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv.*, 46(4). DOI: 10.1145/2542049.

Mohamed, N. (2025). Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms. *Knowledge and Information Systems*, pages 1–87. DOI: 10.1007/s10115-025-02429-y.

Nguyen, H. B., Xue, B., Liu, I., and Zhang, M. (2014). Filter based backward elimination in wrapper based pso for feature selection in classification. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3111–3118. DOI: 10.1109/CEC.2014.6900657.

ORG, S. (2024). Welcome to the SHAP documentation. Available at:`https://shap.readthedocs.io/en/latest/index.html`. 31/01/2025.

Pasupathi, S., Kumar, R., and Pavithra, L. (2025). Proactive DDoS detection: integrating packet marking, traffic analy-

sis, and machine learning for enhanced network security. *Cluster Computing*, 28(3):210. DOI: 10.1007/s10586-024-04849-x.

Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., and O'Sullivan, J. M. (2022). A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics*, 2:927312. DOI: 10.3389/fbinf.2022.927312.

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. DOI: 10.1109/TNSM.2021.3088763.

Quincozes, S. E., Passos, D., Albuquerque, C., Ochi, L. S., and Mossé, D. (2020). GRASP-based feature selection for intrusion detection in cps perception layer. In *2020 4th Conference on Cloud and Internet of Things (CIoT)*, pages 41–48. DOI: 10.1109/CIoT50422.2020.9244207.

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. DOI: 10.1007/s10207-024-00828-w.

Rana, K., Gupta, S., Kaur, G., and Yadav, A. L. (2024). Malware detection in network traffic using machine learning. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 358–362. IEEE. DOI: 10.1109/ICAAIC60222.2024.10575355.

Rostami, M., Berahmand, K., Nasiri, E., and Forouzandeh, S. (2021). Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100:104210. DOI: 10.1016/j.engappai.2021.104210.

Salehpour, A., Balafar, M. A., and Souri, A. (2025). An optimized intrusion detection system for resource-constrained iomt environments: enhancing security through efficient feature selection and classification. *The Journal of Supercomputing*, 81(6):783. DOI: 10.1007/s11227-025-07253-3.

Santhosh Kumar, S. V. N., Selvi, M., and Kannan, A. (2023). A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Computational Intelligence and Neuroscience*, 2023(1):8981988. DOI: 10.1155/2023/8981988.

Santos, M. R., Guedes, A., and Sanchez-Gendriz, I. (2024). Shapley additive explanations (shap) for efficient feature selection in rolling bearing fault diagnosis. *Machine Learning and Knowledge Extraction*, 6(1):316–341. DOI: 10.3390/make6010016.

Scherer, F., Dresch, F., Quincozes, S., Kreutz, D., and Quincozes, V. (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 Informação e de Sistemas Computacionais*, pages 351–366, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbseg.2024.241780.

Singh, R. P., Dash, R., and Mohapatra, R. K. (2025). Unveiling explainability in face anti-spoofing: Hybrid feature extraction with xai-guided feature aggregation. *Pattern Recognition*, page 111905. DOI: 10.1016/j.patcog.2025.111905.

Song, X., Zhang, Y., Zhang, W., He, C., Hu, Y., Wang, J., and Gong, D. (2024). Evolutionary computation for feature selection in classification: A comprehensive survey of solutions, applications and challenges. *Swarm and Evolutionary Computation*, 90:101661. DOI: 10.1016/j.swevo.2024.101661.

Theng, D. and Bhoyar, K. K. (2024). Feature selection techniques for machine learning: a survey of more than two decades of research. *Knowledge and Information Systems*, 66(3):1575–1637. DOI: 10.1016/j.apenergy.2023.122079.

Tran, B., Xue, B., and Zhang, M. (2014). Improved pso for feature selection on high-dimensional datasets. In *Simulated Evolution and Learning: 10th International Conference, SEAL 2014, Dunedin, New Zealand, December 15-18, 2014. Proceedings 10*, pages 503–515. Springer. DOI: 10.1007/978-3-319-13563-2_43.

Van Zyl, C., Ye, X., and Naidoo, R. (2024). Harnessing explainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of grad-cam and shap. *Applied Energy*, 353:122079. DOI: 10.1016/j.apenergy.2023.122079.

Vijayanand, R. and Devaraj, D. (2020). A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network. *IEEE Access*, 8:56847–56854. DOI: 10.1109/ACCESS.2020.2978035.

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. DOI: 10.1016/j.engappai.2022.105591.