

Article

TrustEdge-V2X: Deployment-Aware Edge Intelligence for V2X/IoV Intrusion and Misbehavior Detection

Hesham A. Sakr ^{1,2,3,4,*}, Mina Shenouda ⁵, Nadeem Sarwar ⁶ , Ibrahim Elewah ⁷ , Vitalii Lapin ⁸ 
and Maria Lapina ^{8,9} 

¹ Faculty of Computing and IT, Sohar University, P.O. Box 44, Sohar 311, Oman

² Nile Higher Institute of Engineering and Technology, Mansoura 35511, Egypt

³ Department of Information Technology, College of Informatics, MidOcean University, Moroni RP 98123, Comoros

⁴ Faculty of Engineering Technology, Elsewedy University of Technology (SU Tech), Cairo 44629, Egypt

⁵ Software Engineering Department, Faculty of Information Technology, Philadelphia University, Amman 19392, Jordan; melnour@philadelphia.edu.jo

⁶ Department of Computer Science, Bahria University Lahore Campus, Lahore 54000, Pakistan; nadeem_srwr@yahoo.com

⁷ College of Engineering and Technology, American University of the Middle East, Dasman 15453, Kuwait; ibrahim-elewah@aum.edu.kw

⁸ Department of Computational Mathematics and Cybernetics, North-Caucasus Federal University, 355017 Stavropol, Russia; vitlx@yandex.ru (V.L.); mlapina@ncfu.ru (M.L.)

⁹ Trusted AI Research Center, Russian Academy of Sciences, 109004 Moscow, Russia

* Correspondence: hsakr@su.edu.om

Abstract

Most studies on vehicular cybersecurity focus on classification accuracy but fail to evaluate the applicability of the models to the edge. This paper proposes a secure edge intelligence framework for V2X/IoV intrusion and misbehavior detection, called TrustEdge-V2X, which is deployment-aware. The framework combines dataset-role qualification, attack taxonomy, AI model benchmarking, feature-budget analysis, deployment ranking based on EdgeScore, offline risk-aware orchestration, external validation, robustness testing, explainable AI, repeated-run statistical analysis and ablation studies. Three datasets are assigned different experimental roles: VeReMi_NextGen is used for core V2X/VANET misbehavior detection, CICIoT2023 is used for supporting edge/IoT intrusion experiments and HCRL_CarHacking is used for external IoV/CAN validation. LightGBM outperformed all other AI models in EdgeScore (0.9383), F1-score (0.9878), MCC (0.9758), and inference latency (0.009419 ms per sample) across all eight AI models and six scenarios on VeReMi_NextGen for binary detection. In five dataset-task cases, the accuracy-best model was different from the EdgeScore-best model, which is the most important point to note: the best model in terms of accuracy is not necessarily the best model in terms of EdgeScore. Compact feature subsets were competitive, and robustness testing demonstrated an average F1 decrease of 0.1423 when tested under stress. The orchestration layer was found to be beneficial for the tasks, but it did not always perform better than the best fixed policy. As a whole, TrustEdge-V2X offers a systematic approach to the assessment and selection of vehicular cybersecurity models based on the operational and deployment conditions, not only on the classification accuracy.

Keywords: V2X; IoV; VANET security; intrusion detection; misbehavior detection; edge intelligence; Edge AI; EdgeScore; orchestration; robustness; XAI



Academic Editor: Amit Kumar Mishra

Received: 24 July 2026

Revised: 23 August 2026

Accepted: 25 August 2026

Published: 2 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

The Vehicle-to-Everything (V2X) systems, Internet of Vehicles (IoV) architectures, and in-vehicle Controller Area Network (CAN) subsystems have become a part of the intelligent transportation systems. The advantages are offset, however, by an increased attack surface in the form of false-position messages, replay, Sybil, denial of service, spoofing, in-vehicle command manipulation and more. Recent surveys of V2X and intelligent-vehicle security focus on the fact that the primary challenge is not just to detect malicious traffic, but also to detect it reliably, explainably, and in low latency environments at the edge of the network [1–4].

The vast majority of the literature that has been published states that accuracy, macro-F1 or AUC are the primary indicators of success. While useful, these measurements are not sufficient for the practical deployment of V2X/IoV. If the model size, throughput, model inference latency or energy-related cost are not suitable, the model may be a good candidate for a clean test, but it is not suitable to be used in an on-board unit or roadside unit or fog node. This paper hence considers security performance and deployment feasibility as a single decision problem instead of separate post considerations.

TrustEdge-V2X is tackling this issue by leveraging a deployment-aware and dataset-driven framework. It compares the performance of different binary, family-level and multiclass detection tasks, lightweight, medium and heavy AI models, and ranks the candidates using EdgeScore, evaluates the impact of different features, validates the detection task offline, validates the detection task in the external CAN domain, evaluates the robustness, XAI, repeated-run stability, and ablation. The work is not to claim packet-level VANET routing simulation. The contribution is an offline secure edge intelligence framework, which is based on measured model profiles and dataset-role qualification.

Compared with existing deployment-aware V2X/IoV security studies, which typically report a single deployment metric (e.g., inference latency or model size) alongside detection accuracy, TrustEdge-V2X differs in three respects: (i) it treats dataset-role qualification, detection performance, and deployment cost as a single evaluation protocol rather than as independent post hoc measurements; (ii) it introduces a composite, explicitly weighted and reproducible EdgeScore rather than a qualitative deployment discussion; and (iii) it couples this ranking with an offline risk-aware orchestration layer that is evaluated against fixed deployment policies, which prior deployment-aware IDS studies for V2X/IoV do not provide. This distinction is the central methodological contribution of the proposed framework.

This paper makes the following main contributions:

- To avoid the overclaiming of a dataset, an evaluation framework based on the dataset role is introduced, clearly separating the core V2X/VANET security evidence, the supporting security evidence for edge/IoT devices and external security validation evidence for IoV/CAN devices. This separation improves the scientific defensibility of the experimental design, and ensures that each dataset only supports the claim for which it is suitable.
- The experimental methodology is designed and implemented in six scenarios for assessing TrustEdge-V2X from different complementary perspectives, including AI-based intrusion detection, cross-dataset feasibility, feature-budget analysis, deployment-aware EdgeScore ranking, offline risk-aware orchestration, external IoV/CAN validation, and robustness stress testing under degraded input conditions.
- A new deployment-aware EdgeScore ranking mechanism, which is beyond the accuracy-centric model selection, is proposed. The proposed ranking is based on the detection reliability indicators (F1-score and MCC) and the edge-deployment factors (inference latency, model size, energy proxy, and throughput behavior).

- The risk-aware orchestration layer is designed to assess the potential for switching between lightweight edge, medium edge/fog and heavier fog-oriented configurations. The orchestration results are reported transparently including successful and less successful cases, as well as gains that are observed, which are assumed to be task dependent and not necessarily all positive.
- A post-detection validation package is built-in to increase the reliability and interpretability of the framework. This package provides explainable AI analysis, repeated-run statistical validation, ablation analysis, confusion-matrix inspection and metric-correlation analysis, and provides a more complete evaluation than traditional IDS testing based on a single metric.
- A balanced strategy of results interpretation is applied that not only emphasizes the best configurations, but also the limitations revealed by the experiments, including cross-dataset incompatibility, loss of robustness, and cases where fixed deployment policies outperform the adaptive orchestration layer.

The rest of this paper follows the structure below. In Section 2, the most relevant studies on V2X/IoV cybersecurity, machine-learning based intrusion and misbehavior detection, edge-aware security intelligence, and explainable/robust IDS evaluation are reviewed, and the research gaps that are filled by this work are emphasized. In Section 3, the proposed TrustEdge-V2X framework is presented, which consists of the dataset-role qualification logic, the overall secure edge intelligence workflow, and the relationship between the detection, EdgeScore ranking, orchestration, and validation layers. Section 4 outlines the experimental methodology, which includes preprocessing, task formulation, model tiers, feature modes, evaluation metrics, EdgeScore computation, and six evaluation scenarios. The results of the experiments are reported and discussed in Section 5, which includes detection performance, deployment-aware ranking, feature-budget analysis, offline orchestration, external IoV/CAN validation, robustness testing, XAI, statistical validation, ablation, and confusion-matrix analysis. Section 6 covers the primary threats to validity and limitations such as dataset heterogeneity, offline orchestration assumptions, feature-space incompatibility and robustness constraints. Last but not least, Section 7 summarizes the paper and provides directions for future work in online deployment, real edge profiling and general validation of V2X/IoV.

2. Related Work

The authentication, misbehavior detection, privacy and future studies aspects have been a topic of extensive research in V2X cybersecurity mechanisms [1]. The detection of misbehavior using machine learning in 5G and beyond vehicular networks is also studied, and the selection of datasets, feature engineering, validation protocols and deployment challenges are highlighted [2]. With more recent intelligent-vehicle IDS surveys, it can be seen that deep learning and hybrid learning are able to enhance detection but suffer from issues of interpretability, robustness, and edge-deployment [3,4]. This encourages a manuscript layout that is based on the combination of detection quality with deployment restrictions.

Another key challenge in vehicular cybersecurity studies is the selection of datasets. The design of VeReMi is tailored for the similar evaluation of misbehavior detection in VANETs and thus is the natural core V2X evidence source in this paper [5]. CICIoT2023 is, on the other hand, an attack dataset for IoT in real-time that covers a wide range of attacks and can be used as supporting evidence for edge/IoT security, but not as a VANET-routing dataset [6]. HCRL_CarHacking is based on CAN traffic that has been recorded using the OBD-II interface from a real car and can therefore be used for external evaluation of HCRL traffic (V2X) and not for evaluation of RSU/OBU traffic (V2X).

Recent IDS studies for IoV and AVs show the benefits of ensemble and adaptive models, but many still focus on the predictive performance rather than a deployment profile. Ullah et al. introduced an ensemble-based hybrid model for the detection of IoV attacks and Aljabri et al. introduced a lightweight and adaptive data-driven IDS for autonomous vehicles [7,8]. A recent survey of ensemble learning for multi-class IoV IDS also mentions that the computational cost and the roles of the dataset need to be better reported for model comparison [9]. To address this, TrustEdge-V2X solves this gap by incorporating model tiering, latency, model size, energy proxy and EdgeScore in the selection logic.

Security in vehicles has also become a focus of growing attention, especially the edge-oriented vehicular security. The V2X misbehavior detection in 5G edge networks has been studied in federated learning, while misbehavior detection based on LLM has been recently studied with VeReMi style evidence [10,11]. These studies confirm the need to go beyond the accuracy of the central offline. But they are not a substitute for a clear offline benchmarking system to determine the behavior of the chosen model in the edge, fog, and more heavily deployed scenarios. The orchestration and action-distribution analysis are specifically the role of TrustEdge-V2X.

This theme is echoed by the broader cyber-physical security literature. The above examples of DDoS detection in IoT multi-energy systems, DDoS detection in IoT-driven 6G energy-hub networks, and false-data-injection detection in smart grids demonstrate that the evaluation of cyber-physical AI systems must be based on reliable and operationally meaningful metrics, rather than just raw accuracy [12–14]. These works are used as supporting motivation as they are not V2X datasets, but they support the need for resource-aware AI evaluation in cyber-physical environments. Beyond intrusion and misbehavior detection specifically, the practical deployment constraints considered in this paper are consistent with the broader V2X communications literature: 5G NR V2X connectivity is itself latency- and resource-constrained [15], and infrastructure-assisted V2I/V2V collaboration frameworks explicitly optimize for constrained power and latency budgets in vehicular edge/fog settings [16]. This supports treating latency, model size, and energy proxy as first-class evaluation criteria alongside detection accuracy, rather than as secondary considerations.

Methodologically, this paper is based on existing model families and evaluation concepts. The model inventory that is evaluated in this work is based on classical and boosting models such as Random Forests, Extra Trees, Gradient Boosting, XGBoost, and LightGBM [17–21]. Imbalance-aware metrics are suggested by the following: MCC, ROC analysis and precision-recall analysis. Lastly, feature-importance analysis for reducing black box risk [22,23] is encouraged by the interpretable machine learning and XAI-for-security surveys.

The literature survey is presented in Table 1 with the number of the literature cited in each group having a specific role in the manuscript. This table also helps keep the reference list from being an unorganized collection of papers. The table is broken down into direct V2X evidence, supporting IoT evidence, external CAN evidence, model foundations, metric foundations, and XAI foundations. This organization makes the related-work section more defensible, and guarantees that the referenced documents are consistent with the claims in the paper.

Table 1. Literature positioning summary mapped to the final reference list.

Ref. No.	Theme	Main Contribution	Remaining Gap	Use in This Paper
[1–4]	V2X/IoV security surveys	Define cybersecurity, misbehavior detection, IDS, deep learning, and intelligent-vehicle security challenges.	Often survey-oriented and not tied to one deployment-aware experimental pipeline.	Used to motivate TrustEdge-V2X and frame the security gap.
[5]	VeReMi/VANET dataset	Provides a comparable VANET misbehavior detection dataset.	Does not by itself provide edge profiling, orchestration, robustness, or XAI.	Used as the core V2X/VANET evidence source.
[6]	CICIoT2023 IoT benchmark	Provides broad large-scale IoT attack evidence.	Not a VANET routing or V2X message dataset.	Used only as supporting edge/IoT security evidence.
[24]	HCRL Car-Hacking	Provides real-vehicle CAN traffic with DoS, fuzzy, gear-spoofing, and RPM-spoofing attacks.	Represents in-vehicle CAN behavior, not RSU/OBU V2X communication.	Used as external IoV/CAN validation.
[7–9]	IoV/autonomous-vehicle IDS	Show ensemble, adaptive, and multi-class IoV IDS directions.	Need stronger deployment-cost and dataset-role reporting.	Used to position TrustEdge-V2X as deployment-aware rather than accuracy-only.
[10,11]	Edge/federated/LLM V2X detection	Show emerging edge, federated, and LLM-assisted V2X security directions.	Do not provide the same unified offline EdgeScore/orchestration workflow.	Used to justify the edge-intelligence and orchestration layers.
[12–14]	Cyber-physical AI security	Show resource-aware ML evaluation in related cyber-physical domains.	Supporting context only; not direct V2X evidence.	Used to support the broader deployment-aware evaluation motivation.
[17–21]	Model foundations	Define Random Forest, Extra Trees, Gradient Boosting, XGBoost, and LightGBM.	Algorithmic foundations rather than vehicular-specific studies.	Used to justify the selected model inventory.
[25–27]	Metric foundations	Support MCC, ROC, and PR metrics under class imbalance.	Do not cover edge deployment by themselves.	Used to justify F1/MCC/ROC/PR reporting.
[22,23]	XAI foundations	Support feature-importance and explainability analysis.	XAI does not prove causality by itself.	Used to justify the XAI validation subsection.

3. Proposed TrustEdge-V2X Framework

The entire pipeline is shown, not a single classifier, as in Figure 1. The workflow begins with dataset qualification and attack taxonomy, followed by preprocessing, model training, edge profiling, EdgeScore ranking, and finally the validation layers. It is crucial that the orchestration block follows EdgeScore ranking: routing is based on model profiles measured, not on model selection. This directly answers the question of how the six scenarios relate to the last deployment-aware claim. Practically, Figure 1 highlights that detection performance alone does not determine model suitability: only configurations that pass through the EdgeScore and orchestration layers are treated as deployment-ready, which is the central practical message of the proposed framework.

The evidence boundary is made explicit in Figure 2. The core V2X/VANET misbehavior dataset is VeReMi_NextGen, while edge/IoT evidence is supported by CICIoT2023, and HCRL_CarHacking is external IoV/CAN validation. This division will avoid overclaiming. It guarantees that the V2X claims are primarily based on VeReMi_NextGen, while V2X supporting and external-domain behavior are tested on CICIoT2023 and HCRL_CarHacking. The practical implication of Figure 2 is that any claim of state-of-the-art V2X/IoV performance in this paper is scoped to VeReMi_NextGen; results obtained on CICIoT2023 or HCRL_CarHacking support, but do not replace, this core evidence.

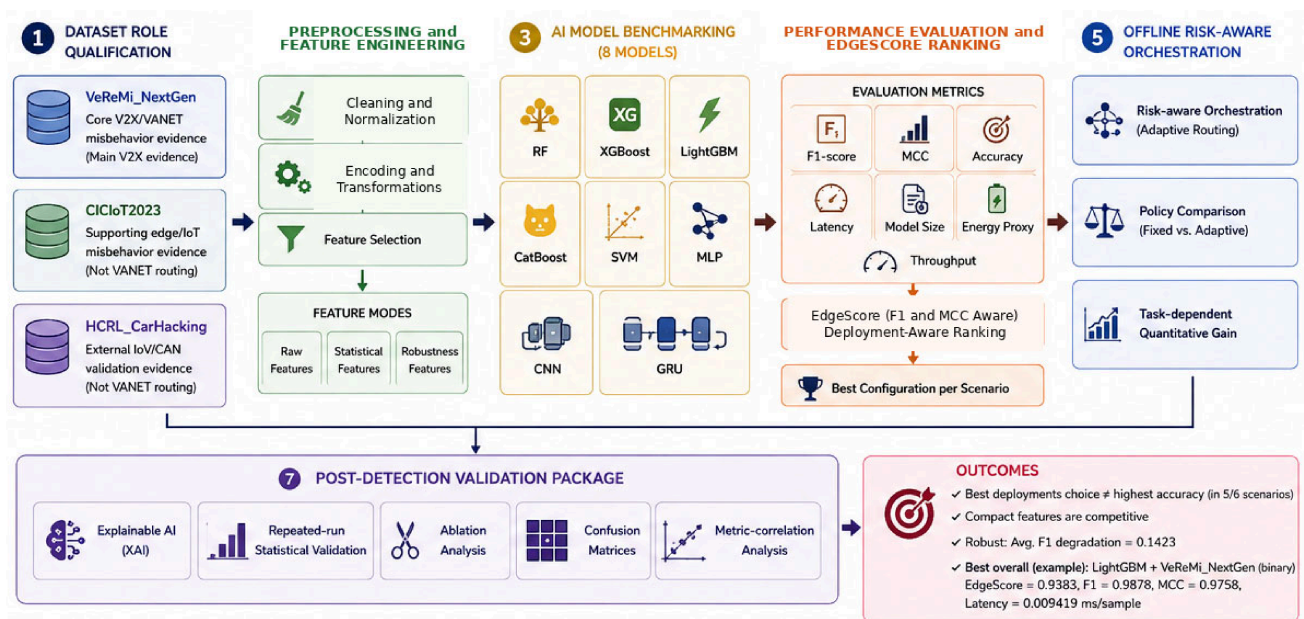


Figure 1. Proposed TrustEdge-V2X workflow integrating dataset qualification, attack taxonomy, AI detection, edge profiling, EdgeScore ranking, orchestration, robustness analysis, XAI, statistical validation, and ablation. Solid arrows indicate the forward processing sequence; dashed arrows indicate feedback or validation links; database symbols denote datasets or stored evidence; model and gear symbols denote learning and deployment operations.



Figure 2. Dataset-role qualification map used in TrustEdge-V2X. Each dataset is mapped to one role—core evidence, supporting evidence, or external validation—indicating how much weight its results carry in the paper’s conclusions.

Dataset Qualification and Exploratory Analysis

Table 2 shows that there are 30,000 rows of data in each of the tables, with varying numbers of columns: CICIOT2023 has 53 columns, HCRL_CarHacking has 12 columns, and VeReMi_NextGen has 37 columns. This indicates that the sets are not interchangeable. The paper-use decision column clarifies the evidential role assigned to each dataset. The datasets of CICIOT2023 and HCRL_CarHacking are not presented as VANET routing datasets, whereas VeReMi_NextGen is reserved as the core V2X/VANET misbehavior evidence. The differentiated experimental roles reflect the intended scope of each dataset rather than an arbitrary assignment. VeReMi_NextGen is the only dataset that captures inter-vehicle V2X/VANET messages with ground-truth misbehavior labels, so it is qualified as the core evidence source for the paper’s central claims. CICIOT2023 captures IoT/edge network traffic rather than inter-vehicle communication; it is retained to test whether the proposed detection and edge-profiling pipeline generalizes to a broader, non-V2X edge-security setting, but it is not used to support V2X-specific claims. HCRL_CarHacking captures in-vehicle CAN-bus traffic, which is architecturally distinct from both inter-vehicle V2X messaging and IoT network traffic; it is therefore used strictly as an external, out-of-domain validation set to test whether the framework’s ranking behavior transfers to a

different IoV attack surface rather than being dataset-specific. This qualification logic is summarized in Figure 2 and Table 2.

Table 2. Qualified datasets, roles, and paper-use decisions.

Dataset	Rows	Columns	Files	Role	V2X Relation	Paper Decision
CICIoT2023	30,000	53	3	EDGE_IOT_SECURITY_SUPPORT_ONLY	IoT/edge security support only	Do not present as VANET routing dataset
HCRL_CarHacking	30,000	12	9	EXTERNAL_IOV_CAN_VALIDATION_ONLY	External IoV/CAN validation only	Do not present as VANET routing dataset
VeReMi_NextGen	30,000	37	1	CORE_V2X_VANET_MISBEHAVIOR	Direct V2X/VANET misbehavior evidence	Use as core dataset

Figure 3 shows high feature redundancy in CICIoT2023. The strongest reported correlations include Srate/Rate = 1.000000, Radius/Std = 0.999984, Weight/Number = 0.999140, and IAT/Weight = 0.997291. These values are used for the following feature-budget analysis. If several features are almost similar, then small feature modes may help to lower the deployment cost without compromising the detection quality, but this will need to be tested empirically. A practical consequence of this multicollinearity is that mutual-information-based top-k feature selection can retain several near-duplicate features rather than k independent sources of evidence; the feature-budget results in Scenario 3 (Section 5.4) should therefore be read as evidence that a smaller feature count is sufficient for competitive detection, not as evidence that the retained features are mutually independent.

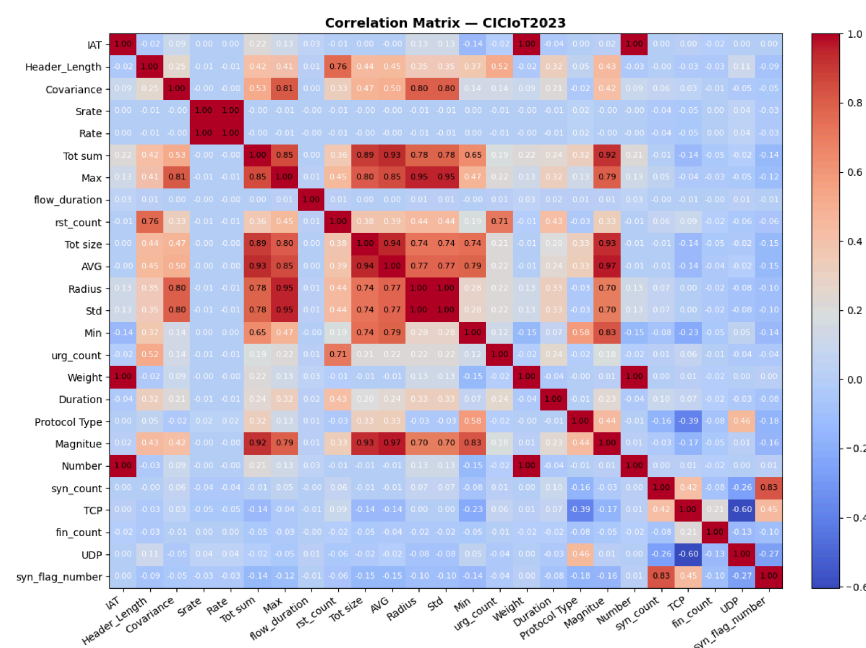


Figure 3. Correlation matrix for CICIoT2023 numeric features. Cell color and value indicate the Pearson correlation between each pair of numeric features, with near-1.0 values flagging redundant, near-duplicate features.

Figure 4 shows the correlation matrix for the HCRL_CarHacking dataset. The results indicate that SubClass and Arbitration_ID are strongly related, meaning that certain CAN identifiers are linked to specific traffic classes. The attack label shows a noticeable relationship with DLC, suggesting that message length may help distinguish malicious traffic. In contrast, Data has weak correlation with most features, indicating that payload information

may require non-linear analysis. Overall, the figure suggests that CAN identifiers, timing, and DLC are important indicators for detecting attacks, while machine-learning models are needed to capture more complex relationships.

Correlation Matrix — HCRL_CarHacking

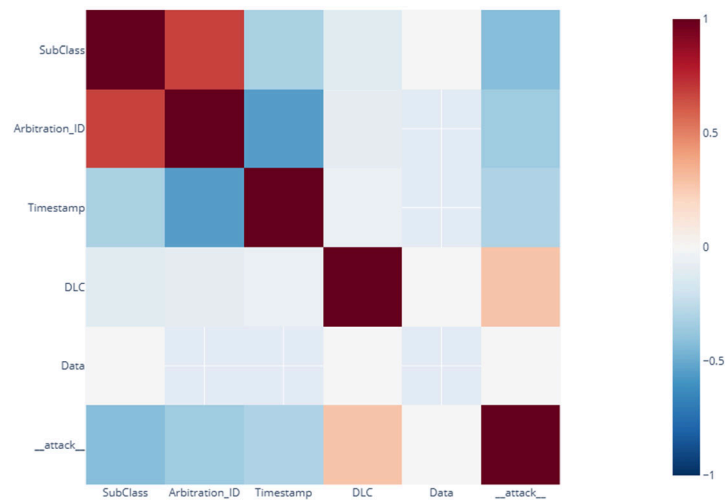


Figure 4. Correlation matrix for HCRL_CarHacking numeric features. Cell color and value indicate the Pearson correlation between each pair of numeric CAN-frame features (e.g., Arbitration_ID, DLC, Data).

Figure 5 shows that VeReMi_NextGen does have high message and mobility correlations. The best pair is messageID/sendTime = 0.998920, next is messageID/sender = 0.958432, and next is sender/sendTime = 0.957986. The following set of values is used to support its use as the central V2X/VANET dataset. The relationships between features are bound to the sender, the time, the location, the heading and the motion behavior, and these are all key aspects of misbehavior detection.

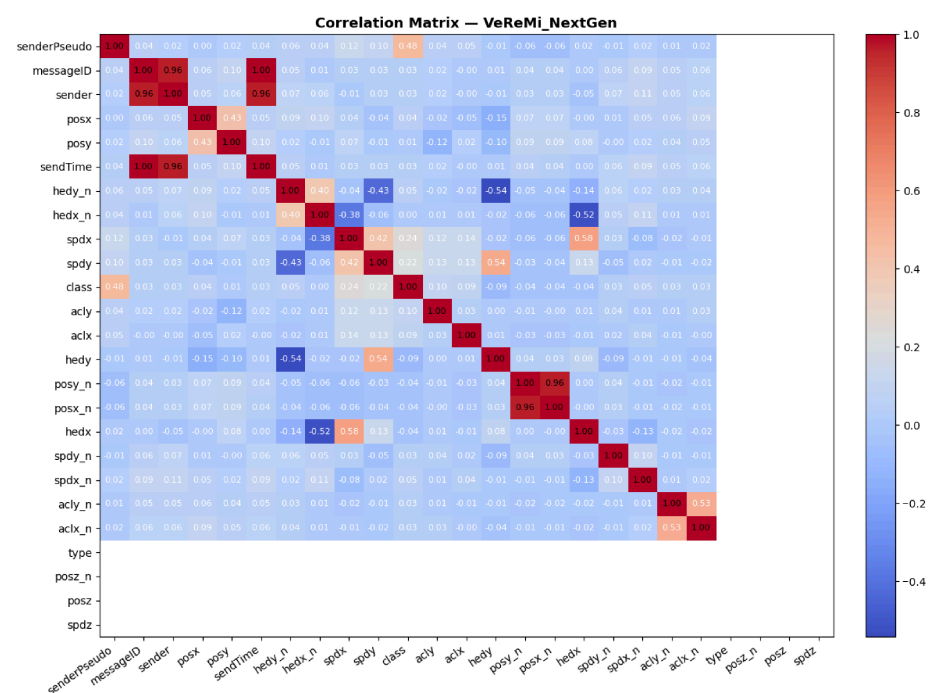


Figure 5. Correlation matrix for VeReMi_NextGen numeric features. Cell color and value indicate the Pearson correlation between each pair of numeric V2X message features (e.g., sender, sendTime, position, heading).

4. Methodology

Software and computational environment: All experiments were executed in Google Colab (Google LLC, Mountain View, CA, USA) using Python 3. The open-source implementation used scikit-learn (Random Forest, Extra Trees, Logistic Regression, MLP, PCA, and feature-selection utilities), XGBoost, LightGBM, pandas, NumPy, SciPy, and Matplotlib, installed through pip using the latest versions available at the time of execution. No proprietary software, commercial instruments, physical laboratory reagents, commercial samples, or commercial cell lines were used in this computational study.

TrustEdge-V2X analyzes multiple model tiers and feature modes in a single analysis workflow. All data is numerically processed, invalid values are dealt with, missing values are added as necessary, and rare and unsuitable classes are filtered as necessary. Classification tasks are binary detection, family-level detection (if attack families are available) and multiclass detection. Feature modes: full features, top-k selections, and PCA-based alternatives. The approach is based on scenarios. In Scenario 1, AI detection performance is reported. In Scenario 2, it is checked whether the data is feasible across the datasets. In Scenario 3, feature budgets and EdgeScore are assessed. Offline risk-aware orchestration is assessed in Scenario 4. In Scenario 5, external IoV/CAN validation is assessed on HCRL_CarHacking. The robustness of the scenario is assessed in the presence of Gaussian noise and feature masking in scenario 6. There are further subsections reporting XAI, repeated run stability, ablation, and confusion matrices. This helps the reviewer to understand the experimental procedure before evaluating the results.

The six scenarios are defined in Table 3 prior to the Section 5, giving the reader an idea of the reasons for the inclusion of each table and figure. The table also makes it clear that it is not a no-load performance test but a feasibility test. This is crucial because under the common-feature threshold it was found that the direct cross-dataset transfer was not meaningful, as shown in the original results.

Table 3. Scenario definitions used in the experimental methodology.

Scenario	Name	Purpose
S1	AI-based detection benchmarking	Compares eight AI models over binary, family, and multiclass tasks using accuracy, balanced accuracy, precision, recall, F1, MCC, latency, throughput, model size, energy proxy, and EdgeScore.
S2	Cross-dataset generalization feasibility	Checks whether dataset pairs share enough common numeric features for meaningful train-test transfer. All six pairings were skipped because fewer than five common numeric features were available.
S3	Feature-budget and EdgeScore analysis	Compares full, top10, top20, and PCA-like feature modes to test whether compact features can preserve security quality while improving deployment suitability.
S4	Offline risk-aware orchestration	Compares fixed policies such as A1 lightweight edge, A2 medium edge/fog, and A3 heavy fog with the proposed adaptive policy using deployment-aware reward logic.
S5	External IoV/CAN validation	Reports HCRL_CarHacking results separately to evaluate vehicular cybersecurity behavior outside the core V2X message dataset.
S6	Robustness stress testing	Applies Gaussian noise and feature masking to measure how clean-test performance degrades under stressed inputs.

4.1. Detection Metrics

The evaluation uses accuracy, balanced accuracy, precision, recall, F1-score, MCC, ROC-AUC, PR-AUC, and false alarm rate (FAR). Macro-F1 and MCC are emphasized

because they are more informative than accuracy under class imbalance. The following equations are written in a stable Word-readable format to avoid broken equation rendering.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2)$$

Precision measures how many predicted attacks are actually attacks. Here, TP and FP are as defined for Equation (1).

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (3)$$

Recall measures how many actual attacks are successfully detected. Here, TP and FN are as defined for Equation (1).

$$\text{F1} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (4)$$

The F1-score summarizes precision and recall into a harmonic mean. Here, Precision and Recall are the quantities computed in Equations (2) and (3).

$$\text{Balanced Accuracy} = (\text{TPR} + \text{TNR}) / 2 \quad (5)$$

Balanced accuracy averages sensitivity and specificity and is useful for imbalanced datasets. Here, TPR (true positive rate) and TNR (true negative rate) denote recall computed on the positive and negative classes, respectively.

$$\text{MCC} = ((\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})) / \sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})} \quad (6)$$

The Matthews correlation coefficient (MCC) measures the quality of binary classification while considering all confusion-matrix terms [17].

4.2. Edge Profiling Metrics

Deployment profiling measures latency per sample, throughput, model size, memory delta, and energy proxy. These metrics are used to determine whether a model is suitable for edge, edge/fog, or fog/cloud execution.

$$\text{Latency_ms} = 1000 \times \text{Inference_Time_seconds} / \text{N_test} \quad (7)$$

Latency per sample is computed from total inference time and the number of test samples. Here, Inference_Time_seconds is the total wall-clock inference time for the test set and N_test is the number of test samples.

$$\text{Throughput} = \text{N_test} / \text{Inference_Time_seconds} \quad (8)$$

Throughput is the number of test samples processed per second. Here, N_test and Inference_Time_seconds are as defined for Equation (7).

$$\text{EnergyProxy} = \text{Latency_ms} \times (1 + \text{ModelSize_MB}) \times (1 + \text{MemoryDelta_MB} / 100) \quad (9)$$

The energy proxy is a relative deployment-cost indicator, not a physical power measurement. It combines normalized latency, size, and memory behavior. Concretely, the energy proxy multiplies the measured per-sample latency by penalty factors for model

size and memory footprint, so that a configuration that is simultaneously slow, large, and memory-intensive receives a disproportionately high (worse) energy proxy; this multiplicative structure is consistent with how latency, memory, and model footprint jointly, rather than additively, constrain constrained-edge devices.

4.3. EdgeScore and Orchestration Reward

EdgeScore is designed to rank models by joint security and deployment suitability. It rewards high F1-score and MCC and rewards deployment-efficient profiles with low latency, small model size, low energy proxy, and high throughput. The exact normalized component values are computed from the experimental profile table. The rationale for combining detection performance with deployment metrics in a single score, rather than reporting them separately, is that in edge-constrained V2X/IoV deployments a detector that is highly accurate but too slow, too large, or too energy-intensive for an onboard unit or roadside unit is not operationally usable regardless of its offline F1-score. Reporting detection and deployment metrics as independent tables leaves this trade-off to the reader; EdgeScore instead fixes an explicit, reproducible weighting scheme (Equation (10)) that gives combined detection quality (F1-score and MCC) the dominant share of the score while still penalizing deployment-infeasible configurations through latency, model size, energy proxy, and throughput.

$$\text{EdgeScore} = 0.35 \times \text{F1_norm} + 0.25 \times \text{MCC_norm} + 0.15 \times \text{LatencyScore} + 0.10 \times \text{SizeScore} + 0.10 \times \text{EnergyScore} + 0.05 \times \text{ThroughputScore} \quad (10)$$

Here, F1_norm is the macro F1-score in its native [0, 1] range; MCC_norm = (MCC + 1)/2 rescales the Matthews correlation coefficient from [−1, 1] to [0, 1]; LatencyScore, SizeScore, and EnergyScore are min-max normalized (1 = best, i.e., lowest latency, smallest size, or lowest energy proxy in the evaluated pool; 0 = worst) from the corresponding raw latency, model-size, and energy-proxy values; and ThroughputScore is min-max normalized in the opposite direction, so the highest-throughput configuration scores 1. All four deployment-related components (LatencyScore, SizeScore, EnergyScore, ThroughputScore) are normalized jointly across the full pool of 224 evaluated model-dataset-task-feature-mode configurations reported in Scenario 1, not separately per dataset or task, so that EdgeScore values remain comparable across the whole study. The weights (0.35, 0.25, 0.15, 0.10, 0.10, 0.05) were fixed a priori so that combined detection quality (F1-score and MCC) accounts for 60% of the score, reflecting that detection reliability remains the primary requirement, while the remaining 40% is distributed across latency, model size, energy proxy, and throughput in decreasing order of their practical relevance to edge deployment. This specific weight configuration, and its influence on the reported rankings, is examined quantitatively in the new Weight Sensitivity Analysis subsection below.

The offline orchestration layer compares fixed policies against a proposed risk-aware policy. Fixed policies always choose a lightweight, medium, or heavy tier, whereas the proposed policy routes decisions according to risk and confidence indicators derived from model profiles.

$$\text{Reward} = \text{lambda1 DetectionUtility} + \text{lambda2 EdgeScore} - \text{lambda3 RiskPenalty} - \text{lambda4 LatencyPenalty} \quad (11)$$

Here, DetectionUtility is the normalized detection-quality term (a weighted combination of F1_norm and MCC_norm, as in EdgeScore); RiskPenalty is a normalized penalty that increases with the estimated misclassification risk of the routed configuration; and LatencyPenalty is the normalized latency cost of the tier selected by the orchestration policy for that sample. The weights lambda1, lambda2, lambda3, and lambda4 set the relative importance of detection utility, deployment suitability (EdgeScore), risk avoidance, and la-

tency control, respectively, in the offline reward signal used to train the orchestration policy; consistent with the EdgeScore weighting rationale (Section 4.3), λ_1 and λ_2 are set so that detection- and deployment-quality terms dominate the reward, while λ_3 and λ_4 discourage high-risk or high-latency routing decisions.

The reward expression summarizes the goal of preserving detection quality while controlling latency and deployment risk. It is used as an offline scoring interpretation rather than as evidence of packet-level VANET routing simulation.

5. Results and Discussion

The overall publication/deployment ranking for the configurations evaluated is summarized in Figure 6. The top-ranked entries are clustered around the VeReMi_NextGen binary task, which is the one case explicitly selected as the benchmark V2X/VANET misbehavior dataset, thus reinforcing the idea that the best evidence for the proposed framework is obtained from this dataset. The optimal fixed detection setup is LightGBM_Heavy_AI on VeReMi_NextGen binary detection with all features, with $F1 = 0.9878$, $MCC = 0.9758$, latency = 0.0094 ms/sample and EdgeScore = 0.9383. This means that the top performing configuration is not only correct, but is also very suitable for edge deployment. The leaderboard also demonstrates the need for more than one metric. There are multiple configurations that can get very close to the same publication scores, but have different latencies, budgets, model complexity and orchestration behavior. VeReMi_NextGen binary results, for instance, are frequently present in the top results as both the detection quality and the deployment profile are good. Other deployments that are more accurate are ranked lower given their less desirable deployment cost or EdgeScore behavior. Hence, the main argument of TrustEdge-V2X is that it considers security models as deployment candidates rather than just offline classifiers, and this is backed by the results shown in Figure 6.

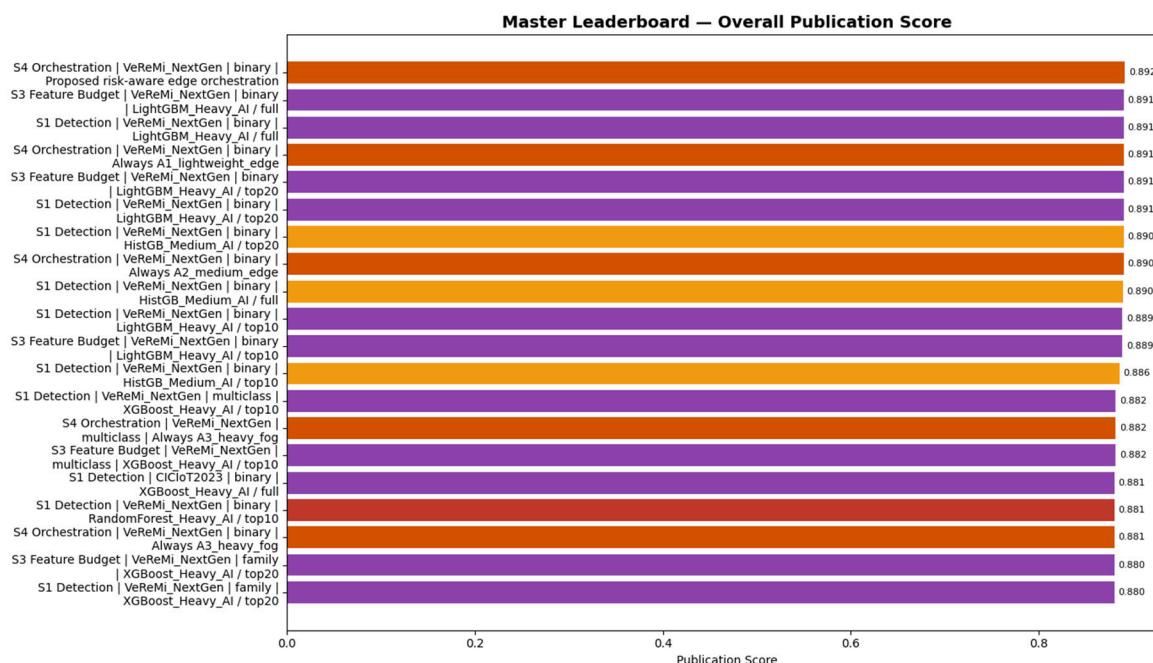


Figure 6. Master leaderboard across the evaluated scenarios. Rows are sorted by EdgeScore in descending order; color intensity encodes the EdgeScore value, allowing the best deployment-aware configuration per scenario to be identified without consulting the underlying tables.

5.1. Scenario 1: AI-Based Detection Performance

The results of the core V2X in Table 4 demonstrate the superiority of VeReMi_NextGen. The best binary VeReMi_NextGen configuration reaches $F1 = 0.9878$, $MCC = 0.9758$,

latency = 0.0094 ms/sample, and EdgeScore = 0.9383. Using XGBoost for VeReMi_NextGen family detection results in F1 = 0.9579 and EdgeScore = 0.9240 with top20 features. The XGBoost with top10 features multiclass detection model for VeReMi_NextGen achieves the following scores: F1 = 0.9822, MCC = 0.9808, and EdgeScore = 0.9283. Random Forest achieves the highest F1-score (0.9543) and EdgeScore (0.9189), while XGBoost is a close second on both metrics (F1 = 0.9503, EdgeScore between 0.918 and 0.924 depending on the measurement run) for CICIOT2023 binary detection, illustrating that these two configurations are effectively tied once latency-measurement variation across runs is taken into account. For CICIOT2023 multiclass detection, XGBoost achieves the highest F1-score (0.8133) and MCC (0.9893). The lower macro-F1 for CICIOT2023 multiclass detection reflects the more difficult fine-grained labels and greater class imbalance of that task compared with binary detection. The quality of HCRL_CarHacking is lower than VeReMi_NextGen: the top LightGBM configurations achieve F1 = 0.7612 and MCC = 0.5323. This is still a useful tool as external IoV/CAN validation but not direct evidence of V2X.

Table 4. Best Scenario 1 detection configurations by dataset and task.

Dataset	Task	Model	Model Tier	Feature Mode	Accuracy	Macro-F1	MCC	Latency (ms/Sample)	Model Size (MB)	Energy Proxy	Edge Score
VeReMi_NextGen	binary	LightGBM_Heavy_AI	Lightweight edge AI model	full	0.9884	0.9878	0.9758	0.0094	0.5316	0.0144	0.9383
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	Heavy fog/cloud AI model	top10	0.9879	0.9822	0.9808	0.0242	3.3517	0.1076	0.9283
VeReMi_NextGen	family	XGBoost_Heavy_AI	Heavy fog/cloud AI model	top20	0.9707	0.9579	0.9515	0.0104	1.7268	0.0283	0.9240
CICIOT2023	binary	RandomForest_Heavy_AI	Heavy fog/cloud AI model	full	0.9959	0.9543	0.9086	0.0077	2.9830	0.0315	0.9189
CICIOT2023	multiclass	XGBoost_Heavy_AI	Heavy fog/cloud AI model	full	0.9903	0.8133	0.9893	0.0283	3.6738	0.1328	0.8680
HCRL_CarHacking	multiclass	LightGBM_Heavy_AI	Lightweight edge AI model	top10	0.9973	0.7612	0.5323	0.0025	0.2118	0.0030	0.8090
HCRL_CarHacking	binary	LightGBM_Heavy_AI	Lightweight edge AI model	top20	0.9973	0.7612	0.5323	0.0025	0.2118	0.0030	0.8090

From the model-comparison perspective the same ranking pattern for Scenario 1 is shown in Figure 7. The lighter cells represent more compelling candidates as regards deployment awareness, and the VeReMi_NextGen binary and family rows continue to be strong areas. Notice that the heatmap also indicates that no one model is best for all data/tasks combinations. This aligns with the choice of a framework-level model selection strategy instead of choosing a model family a priori. The practical significance of Figure 7 is that it identifies, at a glance, which model-dataset-task combinations are deployment-competitive rather than merely detection-competitive, which is the information an integrator would need when selecting a model for an onboard or roadside unit.

Figure 8 is used to complement the EdgeScore heatmap and is used to look at the quality of the detections. The comparison explains why there are some situations where an F1 with a high value is not the best combination in EdgeScore, taking into account latency, model size and energy proxy. One of the most obvious differences between the F1 and EdgeScore heatmaps is the difference in interpretation when the ranking changes during deployment.

5.2. Accuracy-Best Versus EdgeScore-Best Selection

The accuracy-only decision differs from the deployment-aware decision in the locations listed in Table 5. For CICIOT2023 binary and multiclass task, both criteria choose the same models. The choices may differ in HCRL_CarHacking and VeReMi_NextGen. The best model for VeReMi_NextGen multiclass detection is LightGBM_Heavy_AI with accuracy 0.9952 and EdgeScore 0.9075, while XGBoost_Heavy_AI with EdgeScore 0.9283 is the best model for EdgeScore. There is only 0.0073 loss of accuracy with an EdgeScore gain

of 0.0846. The EdgeScore gain for VeReMi_NextGen family detection is 0.0247. The cases illustrate how the central argument that highest accuracy is not always the best deployment decision holds true.

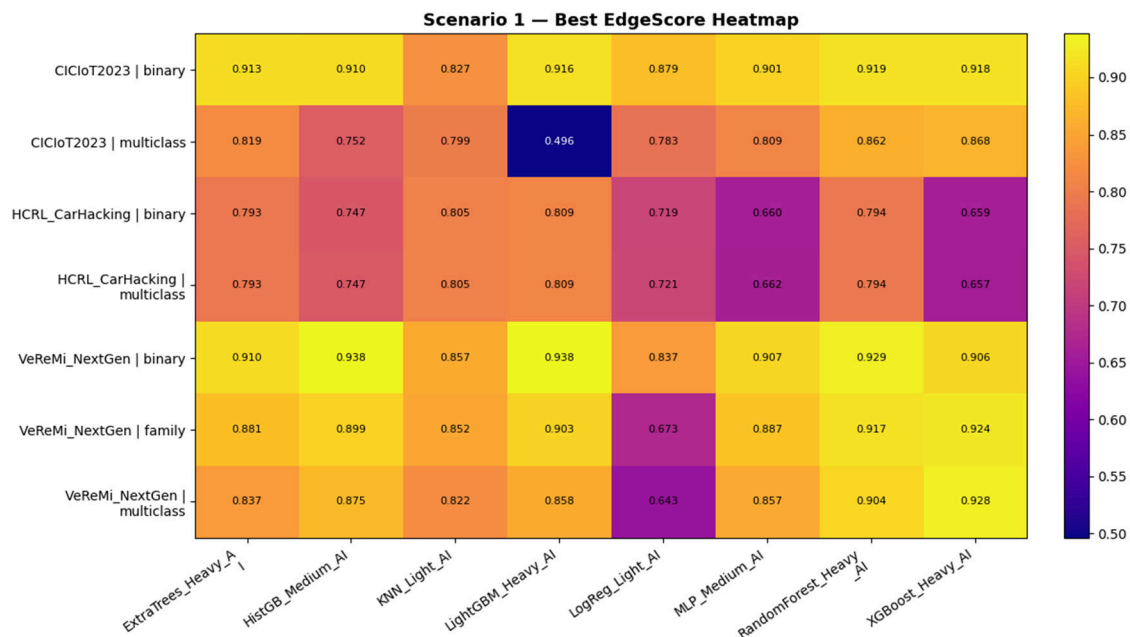


Figure 7. Scenario 1 EdgeScore heatmap across datasets, tasks, and models. Lighter/yellower cells indicate higher values, whereas darker cells indicate lower values; rows correspond to dataset/task combinations and columns to AI models, so a horizontal scan identifies the best-performing model for a given task and a vertical scan identifies the model that is most consistently strong across tasks.

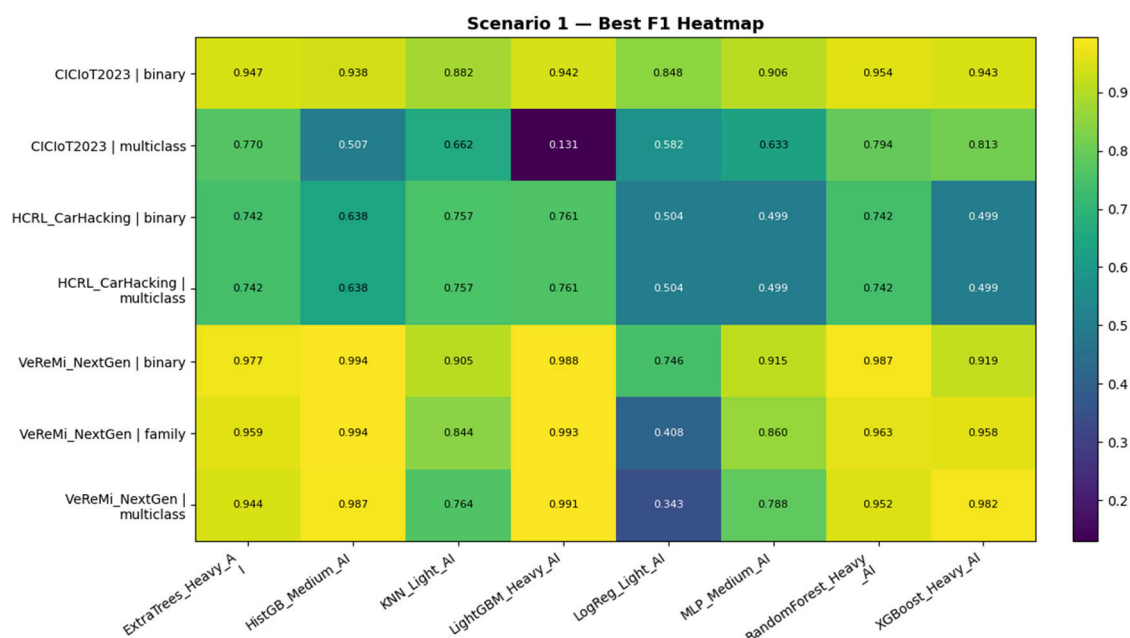


Figure 8. Scenario 1 F1-score heatmap across datasets, tasks, and models. Lighter/yellower cells indicate higher values, whereas darker cells indicate lower values; comparing this heatmap with Figure 7 shows where the F1-best and EdgeScore-best models diverge.

The difference between choosing the best model based on accuracy and choosing the best model based on EdgeScore is visualized in Figure 9. In binary and multiclass tasks (as is the case for CICIoT2023), both criteria choose the same model, implying that the deployment-aware ranking does not influence the model selection in these two scenarios.

The difference becomes apparent, however, in HCRL_CarHacking and VeReMi_NextGen tasks, in which the EdgeScore-best configuration offers a better deployment-aware balance than the accuracy-best configuration. The most significant gains are seen in the VeReMi_NextGen multiclass task with the EdgeScore-based selection score showing an increase of 0.0846 over the accuracy-only selection score, and the accuracy decreasing by only 0.0073. This is an important result because it demonstrates that a slight reduction in the raw accuracy can lead to a significant improvement in the deployment-aware configuration. The same trend is seen for the VeReMi_NextGen family task with an EdgeScore gain of 0.0247. The HCRL_CarHacking gains are slightly smaller, but still positive: 0.0084 for the binary task and 0.0085 for the multiclass task. In short, these results provide direct evidence for the notion that the most accurate model may not be the best model for deployment on the edge. This divergence is practically significant because it shows that selecting a model by accuracy alone leads to a deployment-suboptimal choice in five of the seven evaluated dataset/task cases, reinforcing the need for a deployment-aware ranking such as EdgeScore rather than accuracy alone.

Table 5. Accuracy-best versus EdgeScore-best configuration comparison.

Dataset	Task	Accuracy-Best Model	Accuracy-Best Feature	Accuracy-Best Accuracy	Edge-Best Model	Edge-Best Feature	Edge-Best Edge Score	Edge Gain	Accuracy Loss If Edge Selected
CICIoT2023	binary	RandomForest_Heavy_AI	full	0.9959	RandomForest_Heavy_AI	full	0.9189	0.0000	0.0000
CICIoT2023	multiclass	XGBoost_Heavy_AI	full	0.9903	XGBoost_Heavy_AI	full	0.8680	0.0000	0.0000
HCRL_CarHacking	binary	KNN_Light_AI	top10	0.9977	LightGBM_Heavy_AI	top20	0.8090	0.0084	0.0004
HCRL_CarHacking	multiclass	KNN_Light_AI	top10	0.9977	LightGBM_Heavy_AI	top10	0.8090	0.0085	0.0004
VeReMi_NextGen	binary	HistGB_Medium_AI	full	0.9941	LightGBM_Heavy_AI	full	0.9383	0.0005	0.0057
VeReMi_NextGen	family	HistGB_Medium_AI	top20	0.9971	XGBoost_Heavy_AI	top20	0.9240	0.0247	0.0264
VeReMi_NextGen	multiclass	LightGBM_Heavy_AI	top10	0.9952	XGBoost_Heavy_AI	top10	0.9283	0.0846	0.0073

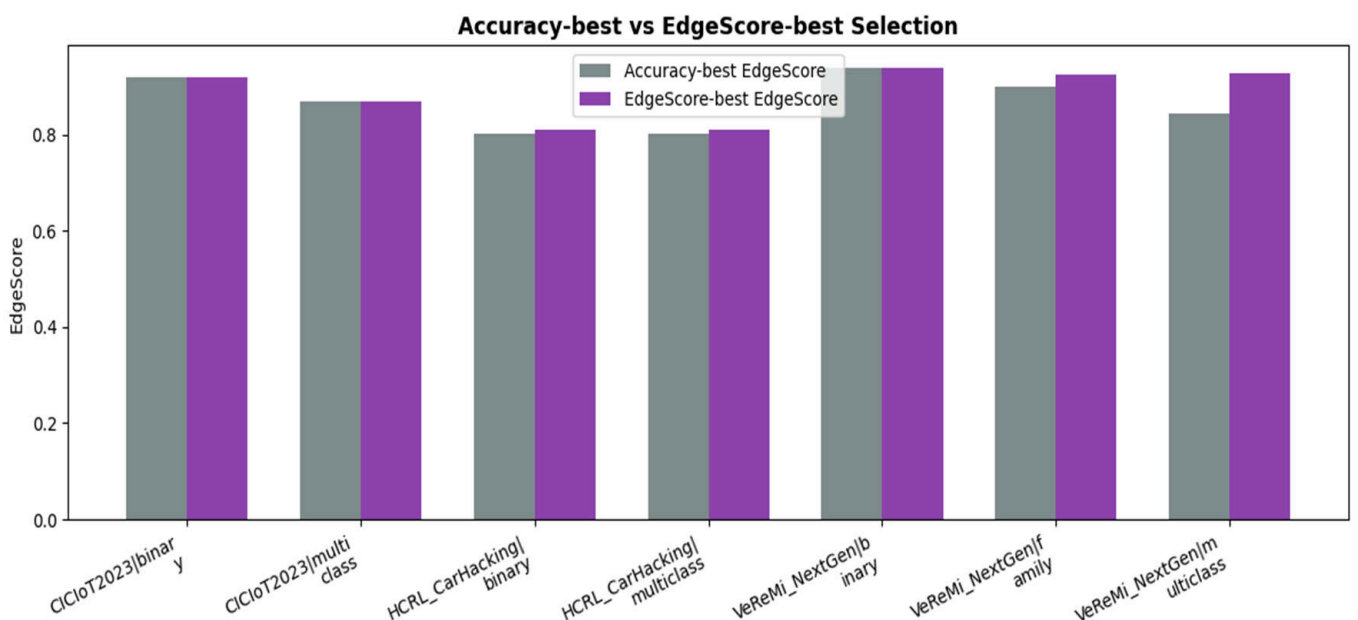


Figure 9. Visual comparison between accuracy-best and EdgeScore-best selections. Paired bars per dataset/task contrast the accuracy-best and EdgeScore-best model choices, so any visible gap between the two bars in a given pair identifies a case where optimizing for raw accuracy would select a different, less deployment-suitable model.

Figure 10 shows the trade-off between the detection quality and inference latency. The optimum region is the top-left corner, where configurations have high F1-scores and low latency. The VeReMi_NextGen binary LightGBM configuration is close to this region, with

latency = 0.0094 ms/sample and F1 = 0.9878. This gives it one of the best deployments due to its high detection quality and low inference delay. The figure also reveals that some high-F1 configurations are more appealing for deployment than others. For instance, VeReMi_NextGen multiclass detection with XGBoost has an excellent F1 = 0.9822, but its latency is raised to 0.0242 ms/sample. Likewise, the multiclass detection result of CICIOT2023 is F1 = 0.8133, with the latency = 0.0283 ms/sample, which is slower and less accurate than the best VeReMi configurations. The HCRL_CarHacking is fast with LightGBM latency of 0.0025 ms/sample, but the F1-score is lower at 0.7612. So as can be seen from Figure 10, low latency is not sufficient, and high F1 is not sufficient. Both requirements must be met for the preferred configuration.

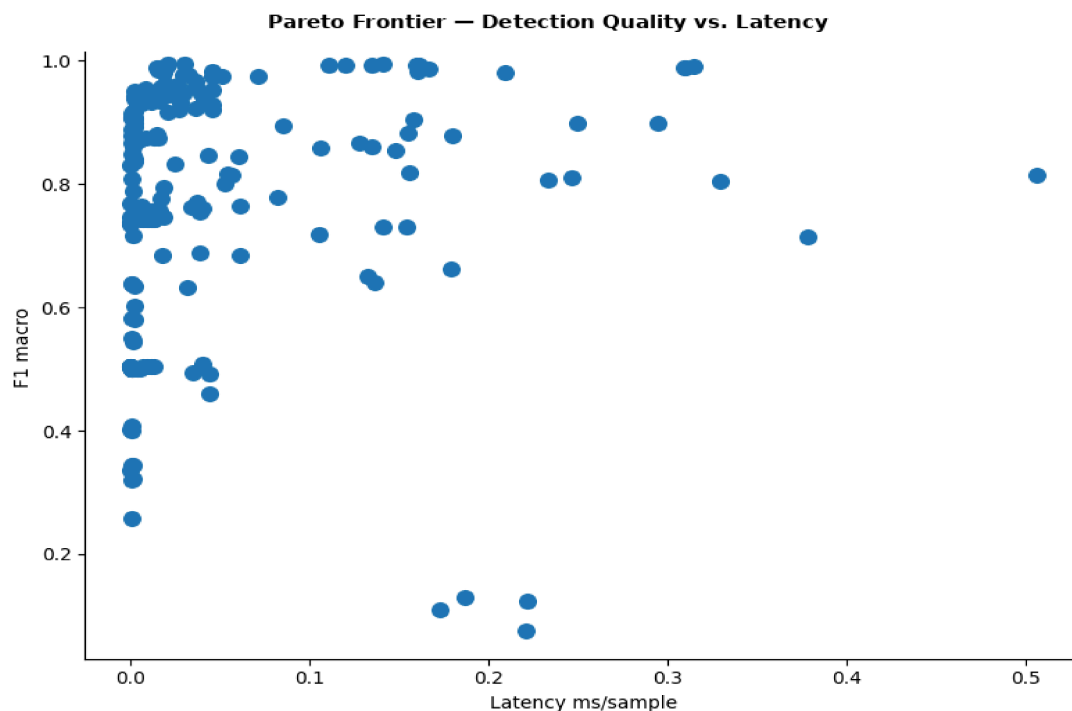


Figure 10. Pareto analysis of F1-score versus latency. Each point is one evaluated configuration; configurations toward the upper-left corner (high F1-score, low latency) approach the Pareto-optimal frontier, where no other configuration achieves both a higher F1-score and a lower latency.

The Pareto analysis is then further extended by replacing latency with the energy proxy in Figure 11. This is important because, although a model might be sufficiently fast for use at the edge, it might not be so appealing if the estimated energy cost is high. The VeReMi_NextGen binary LightGBM configuration once again shows a good balance with F1 = 0.9878 and energy proxy = 0.0144. This shows that the same setup is not only latency optimum, but also energy efficient as well. The energy trade-off is more apparent when comparing heavier models. The performance of VeReMi_NextGen multiclass XGBoost is F1 = 0.9822, and the energy proxy becomes 0.1076. The multiclass XGBoost model has a higher energy proxy (0.1328) for CICIOT2023 model with F1 = 0.8133. In contrast, the energy proxy of HCRL_CarHacking LightGBM is very low (around 0.0030), while the F1-score is only 0.7612. The results demonstrate that the energy efficiency should be considered in combination with the detection reliability. The energy component is thus a good reason to include it in EdgeScore instead of considering it as a secondary metric.

5.3. Scenario 2: Cross-Dataset Generalization Feasibility

In Scenario 2, this study tests the transferability of a classifier trained on one dataset to another dataset, after harmonizing heterogeneous feature spaces. This is a more strin-

gent test than within-dataset testing since the paired datasets do not share a common raw numeric feature intersection. In each of the eight train-test-model combinations, `raw_common_numeric_features = 0`, which would otherwise result in a failure to produce a valid input to a classifier using a simple feature-intersection protocol. In this experiment, instead, each dataset is encoded in a common 75-dimensional semantic-statistical fingerprint before training is conducted on the source dataset and testing is performed on the target dataset, with the name Semantic Statistical Bridge + CORAL. The full results of Scenario 2 calculations are provided in Table 6. All `valid_classifier_output` rows are included in the table, including the best train-test pair. That is significant because the apparent F1 is the value that needs to be interpreted together with the values of balanced accuracy, MCC, FPR and values of the confusion matrix.

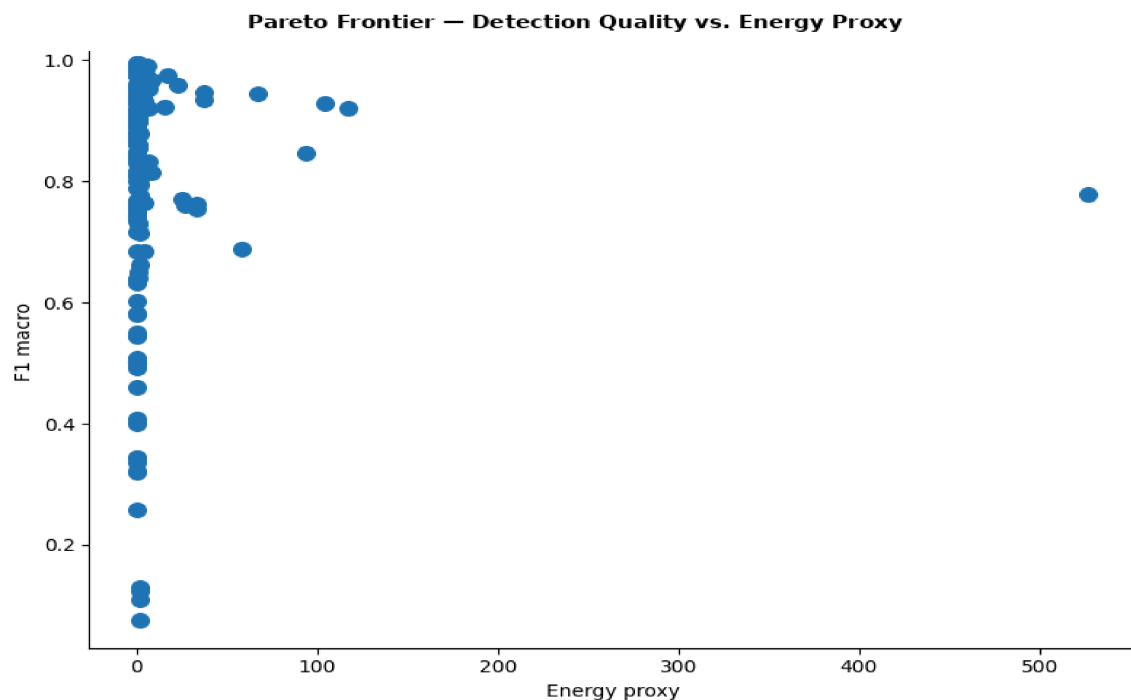


Figure 11. Pareto analysis of F1-score versus energy proxy. Each point is one evaluated configuration; configurations toward the upper-left corner (high F1-score, low energy proxy) approach the Pareto-optimal frontier, where no other configuration achieves both a higher F1-score and a lower energy proxy.

The results first validate the revised Scenario 2 protocol to obtain valid classifiers' output in all eight experiments where no raw feature overlap occurs. The bridge is a translation that maps the VeReMi_NextGen source space (20 numeric features) and the CICIOT2023 source space (40 numeric features) to the same fingerprint space (75 dimensions). Thus, there are no pairs of trains that are not tested. The results of these numbers, however, also highlight the fact that outputs of the two datasets are not strong evidence for deployment-level generalization, but are feasibility evidence. In the VeReMi_NextGen $\rightarrow$ CICIOT2023 direction, Logistic Regression is the most accurate prediction model based on F1. It achieves accuracy = 0.5228, precision = 0.9747, recall = 0.5251, F1 = 0.6825, PR-AUC = 0.9697, and the lowest inference cost in this direction at 0.000136 ms per sample. However, this performance is not strong as measured by more powerful methods: Balanced accuracy = 0.4764, MCC = -0.0143 , ROC-AUC = 0.4414, FPR = 0.5722. The confusion matrix (TN = 299, FP = 400, FN = 13,916, TP = 15,385) implies that almost half of the target attacks are still missed. The highly attack-dominated CICIOT2023 target distribution (with target attack ratio 0.9767) has a significant impact on the high precision and PR-AUC.

Table 6. Complete Scenario 2 cross-dataset harmonized transfer results using Semantic Statistical Bridge + CORAL.

#	Train	Test	Model	RawF	SrcF	TgtF	FPF	TrainN	TestN	S- Atk	T- Atk	Train (s)	Infer (s)	ms/ Smp	Acc	Prec	Rec	F1	BAcc	MCC	TN	FP	FN	TP	FPR	ROC	PR
0	VN	CI	LR	0	20	40	75	30,000	30,000	0.4177	0.9767	2.880	0.0041	0.0001	0.5228	0.9747	0.5251	0.6825	0.4764	−0.014	299	400	13,916	15,385	0.5722	0.4414	0.9697
1	CI	VN	HGB	0	40	20	75	60,000	30,000	0.9767	0.4177	2.482	0.1157	0.0039	0.4177	0.4177	1.0000	0.5893	0.5000	0.000	0	17,469	0	12,531	1.0000	0.4224	0.3735
2	CI	VN	RF	0	40	20	75	60,000	30,000	0.9767	0.4177	76.945	0.1860	0.0062	0.4177	0.4177	1.0000	0.5893	0.5000	0.000	0	17,469	0	12,531	1.0000	0.4056	0.3681
3	CI	VN	ET	0	40	20	75	60,000	30,000	0.9767	0.4177	10.884	0.3065	0.0102	0.4177	0.4177	1.0000	0.5893	0.5000	0.000	0	17,469	0	12,531	1.0000	0.3962	0.3581
4	CI	VN	LR	0	40	20	75	60,000	30,000	0.9767	0.4177	4.914	0.0042	0.0001	0.3985	0.4063	0.9537	0.5698	0.4769	−0.165	3	17,466	580	11,951	0.9998	0.2952	0.3231
5	VN	CI	RF	0	20	40	75	30,000	30,000	0.4177	0.9767	63.841	0.4402	0.0147	0.3352	0.9350	0.3432	0.5021	0.1716	−0.207	0	699	19,246	10,055	1.0000	0.0101	0.9162
6	VN	CI	HGB	0	20	40	75	30,000	30,000	0.4177	0.9767	4.833	0.8416	0.0281	0.2821	0.9238	0.2888	0.4401	0.1451	−0.232	1	698	20,838	8463	0.9986	0.0143	0.9168
7	VN	CI	ET	0	20	40	75	30,000	30,000	0.4177	0.9767	11.242	0.6154	0.0205	0.2623	0.9184	0.2686	0.4156	0.1343	−0.244	0	699	21,432	7869	1.0000	0.0088	0.9147

The same direction of transfer becomes poorer for the tree-based models. RandomForest_balanced decreases to $F1 = 0.5021$, balanced accuracy = 0.1716, $MCC = -0.2066$, and $ROC-AUC = 0.0101$, with $TN = 0$, $FP = 699$, $FN = 19,246$, and $TP = 10,055$. HistGB further drops to $F1 = 0.4401$, balanced accuracy = 0.1451, $MCC = -0.2325$, and $ROC-AUC = 0.0143$. ExtraTrees_balanced is the weakest in this direction, with $F1 = 0.4156$, balanced accuracy = 0.1343, $MCC = -0.2443$, $ROC-AUC = 0.0088$, and $FPR = 1.0000$. The values are not indicative of helpful transferable discrimination, but rather high rates of target-domain misalignment. In the case of CICIOT2023 -> VeReMi_NextGen, all of the models (HistGB, RandomForest_balanced, and ExtraTrees_balanced) have the same $F1 = 0.5893$, accuracy = 0.4177, recall = 1.0000, balanced accuracy = 0.5000, and $MCC = 0.0000$. The confusion matrices are identical for the three models: $TN = 0$, $FP = 17,469$, $FN = 0$, and $TP = 12,531$. This implies that all the target samples are classified as attack. As a result, the $F1$ value is not a measure of meaningful transfer; but rather it is a prevalence-driven result due to the VeReMi_NextGen target attack ratio of 0.4177. HistGB, RandomForest_balanced, and ExtraTrees_balanced have also low-to-moderate PR-AUC values (0.3735, 0.3681, and 0.3581, respectively), and their ROC-AUC values are below 0.5 (0.4224, 0.4056, and 0.3962, respectively). Logistic Regression in the CICIOT2023 -> VeReMi_NextGen direction is only slightly less extreme in terms of its output, as it is all positive, but not quite all positive. It records $TN = 3$, $FP = 17,466$, $FN = 580$, and $TP = 11,951$, resulting in accuracy = 0.3985, precision = 0.4063, recall = 0.9537, $F1 = 0.5698$, balanced accuracy = 0.4769, $MCC = -0.1647$, $FPR = 0.9998$, $ROC-AUC = 0.2952$, and PR-AUC = 0.3231. It has a very low latency of inference (0.000139 ms per sample), but its FPR is close to the maximum, so it cannot be deployed as a detector between different datasets.

The same conclusion can be drawn from efficiency results. Both the training times of Logistic Regression are modest (2.880 s for VeReMi_NextGen -> CICIOT2023 and 4.914 s for CICIOT2023 -> VeReMi_NextGen), and the fastest at inference in both directions (0.000136 and 0.000139 ms per sample). The training times for RandomForest_balanced are the longest (63.841 s and 76.945 s), but do not provide better cross-dataset reliability. So, increasing model complexity is not a solution to the feature-space and distribution-shift problem in Scenario 2. Overall, Scenario 2 is not meant to be a good performance for cross-dataset evaluation with zero common raw numeric features, but rather a test of the feasibility of cross-dataset evaluation. The low MCC values (-0.2443 to 0.0000), near or below chance balanced accuracy (0.1343 to 0.5000), and extreme FPR values (0.5722 to 1.0000) indicate significant domain shift between the vehicular V2X and IoT traffic spaces. This fortifies the paper by openly telling readers about the meaningful yet challenging results of the cross-dataset analysis and also by setting the motivation for further feature-budget, robustness, drift-aware, and orchestration-aware analyses. The best $F1$ for VeReMi_NextGen -> CICIOT2023 is obtained by LogReg_balanced ($F1 = 0.6825$, $MCC = -0.0143$, $BAcc = 0.4764$, $FPR = 0.5722$). The best $F1$ for CICIOT2023 -> VeReMi_NextGen is achieved by HistGB/RandomForest_balanced/ExtraTrees_balanced with $F1 = 0.5893$; however, the reason is that all of the target predictions are positive ($TN = 0$, $FP = 17,469$, $FN = 0$, $TP = 12,531$), which is why HistGB is selected only as a tie-efficient representative with the shortest training time among the tied tree models (2.482 s).

5.4. Scenario 3: Feature-Budget and EdgeScore Analysis

As expected, the feature mode that wins varies depending on the metric and the dataset (see Table 7). $F1$ or EdgeScore is often the winner for full features, edgeScore is often the winner for compact top-k or PCA-like modes, and latency and energy proxy are often the winners for compact top-k or PCA-like modes. For CICIOT2023 binary detection, the full feature set with RandomForest_Heavy_AI achieves both the highest $F1$ -score (0.9543) and the highest EdgeScore (0.9189), narrowly ahead of XGBoost_Heavy_AI on full features;

the PCA95 variant with Logistic Regression achieves the best latency and energy proxy, at approximately 0.0002. This is an example of the compromise between detection quality and extreme efficiency. LightGBM top10/top20 is the best performer for HCRL_CarHacking and Logistic Regression with full features is the best performer for latency and energy proxy. This is once again the need for feature-budget selection to be coupled with deployment objectives.

Table 7. Feature-mode winners across selected metrics in Scenario 3.

Dataset	Task	Metric	Winning Feature Mode	Winning Model	Value	Direction
CICIoT2023	binary	f1_macro	full	RandomForest_Heavy_AI	0.9543	higher is better
CICIoT2023	binary	edge_score	full	RandomForest_Heavy_AI	0.9189	higher is better
CICIoT2023	binary	latency_ms_per_sample	pca95	LogReg_Light_AI	0.0002	lower is better
CICIoT2023	binary	energy_proxy	pca95	LogReg_Light_AI	0.0002	lower is better
CICIoT2023	multiclass	f1_macro	full	XGBoost_Heavy_AI	0.8133	higher is better
CICIoT2023	multiclass	edge_score	full	XGBoost_Heavy_AI	0.8680	higher is better
CICIoT2023	multiclass	latency_ms_per_sample	top10	LogReg_Light_AI	0.0004	lower is better
CICIoT2023	multiclass	energy_proxy	top10	LogReg_Light_AI	0.0004	lower is better
HCRL_CarHacking	binary	f1_macro	top10	LightGBM_Heavy_AI	0.7612	higher is better
HCRL_CarHacking	binary	edge_score	top20	LightGBM_Heavy_AI	0.8090	higher is better
HCRL_CarHacking	binary	latency_ms_per_sample	full	LogReg_Light_AI	0.0001	lower is better
HCRL_CarHacking	binary	energy_proxy	full	LogReg_Light_AI	0.0001	lower is better
HCRL_CarHacking	multiclass	f1_macro	top10	LightGBM_Heavy_AI	0.7612	higher is better
HCRL_CarHacking	multiclass	edge_score	top10	LightGBM_Heavy_AI	0.8090	higher is better
HCRL_CarHacking	multiclass	latency_ms_per_sample	full	LogReg_Light_AI	0.0001	lower is better
HCRL_CarHacking	multiclass	energy_proxy	full	LogReg_Light_AI	0.0001	lower is better
VeReMi_NextGen	binary	f1_macro	full	HistGB_Medium_AI	0.9939	higher is better
VeReMi_NextGen	binary	edge_score	full	LightGBM_Heavy_AI	0.9383	higher is better

Figure 12 shows a comparison of the impact of various feature budgets on deployment-aware performance. The results demonstrate that there is not a consistent effect from feature reduction across the datasets. Compact feature modes, e.g., top10, top20 or PCA-like, give better latency or energy behavior in some cases, while in others the full feature set is the best choice. The detection result of the full feature set is the highest detection result ($F1 = 0.9543$, EdgeScore 0.9189) for the binary detection of the CICIoT2023 dataset using the model RandomForest_Heavy_AI, with XGBoost_Heavy_AI a close second on both metrics. The best latency and energy performance for the same task is achieved with LogReg_Light_AI + PCA-like mode with the value of around 0.000162 ms/sample.

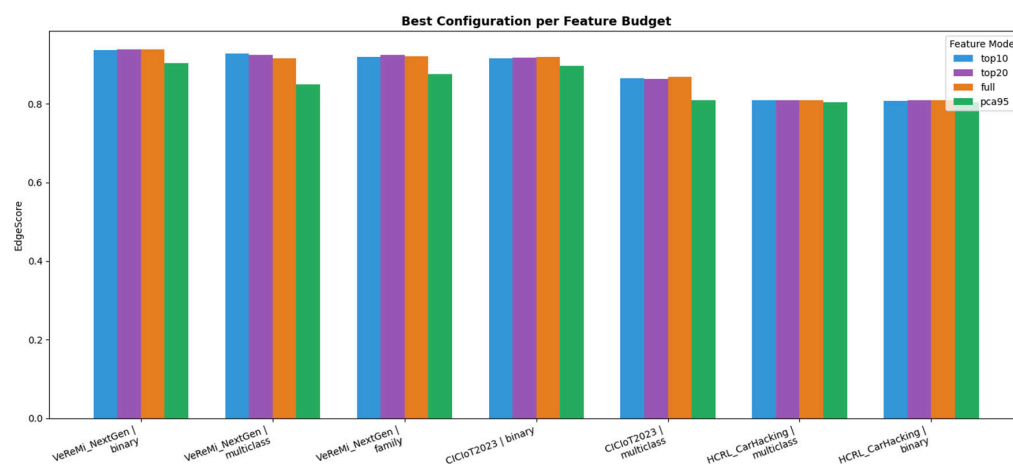


Figure 12. EdgeScore comparison across feature modes. Grouped bars show the best EdgeScore obtained for each dataset/task under full, top10, top20, and PCA95 feature modes.

The feature-budget effect is more complex for HCRL_CarHacking. On the binary task, LightGBM with top10 features gives the best F1-score, and LightGBM with top20 features gives the best EdgeScore, which is 0.8090. That is a sign that the smallest subset of features may not be the best choice for deployment, as sometimes a slightly larger subset of features may be more useful in terms of detection versus deployment behavior. The binary full-feature LightGBM configuration is still the most competitive one overall, with EdgeScore = 0.9383, while compact feature modes are competitive with EdgeScore = 0.9288 for VeReMi_NextGen. So, as shown in Figure 12, feature-budget evaluation is needed instead of assuming that any reduction in features will increase deployment quality.

The results of the evaluation of the feature modes are summarized in Figure 13. As can be seen in the figure, the optimal winning feature mode is quite sensitive to the metric one is optimizing. Even if all feature representation is not preserved in the system, it can be crucial for maintaining the detection reliability, as demonstrated by the F1-score and EdgeScore cases. When using a proxy for latency or energy, however, top10 and top20 and PCA-like modes seem to be more common. This indicates that the feature selection problem is a multi-objective problem. Full features might still be the preferred alternative if the sole objective is raw detection quality. Compact modes might be more appealing in the case of low-latency or low-energy deployment. Practical implication: Selecting features before evaluating the model should be avoided, and should be evaluated with the classifier and the deployment metric. That is why TrustEdge-V2X puts feature-budget analysis into EdgeScore, instead of reducing features as a preprocessing detail.

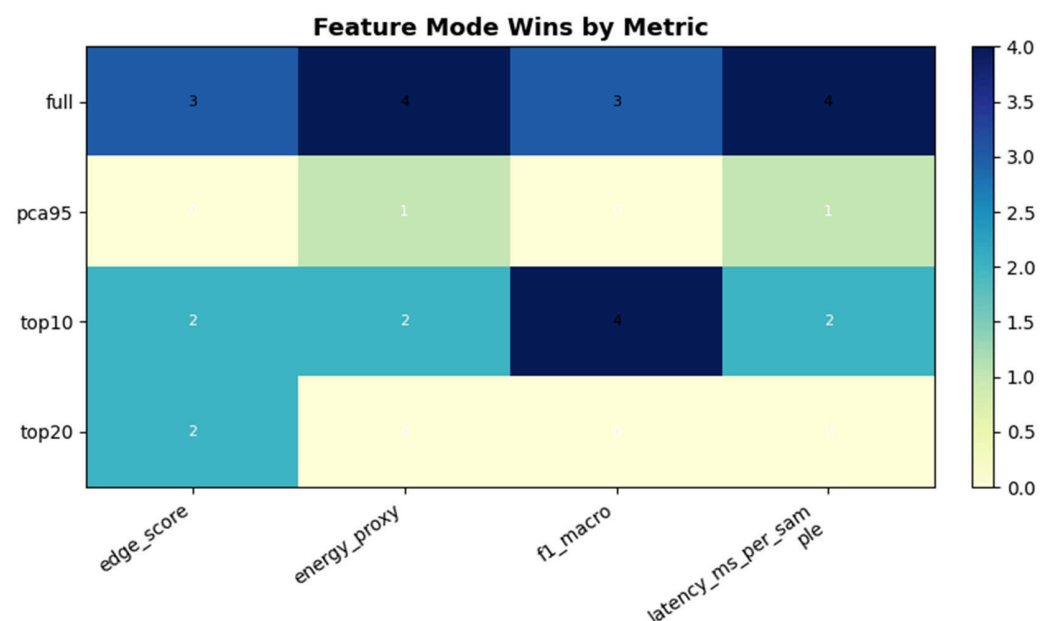


Figure 13. Feature-mode wins by evaluation metric. Each cell reports the number of dataset/task cases in which the corresponding feature mode achieves the best value for that metric.

5.5. Scenario 4: Offline Risk-Aware Orchestration

The key comparison of orchestration is included in Table 8. The only positive gain in CICIOT2023 multiclass is in EdgeScore, which goes from 0.8360 to 0.8364. In all other instances, a better fixed policy is stronger. The proposed policy has EdgeScore = 0.6476 for VeReMi_NextGen multiclass, compared to 0.9005 for the best fixed A3-heavy policy; the difference is -0.2528 . The difference for VeReMi_NextGen family is -0.0801 . These values are significant because they will help to avoid overclaiming. The correct interpretation is that orchestration offers insight into adaptive deployment, but it does not necessarily make the best fixed policy “numerically better.” This is a balanced interpretation that is

more believable for review. Although the proposed orchestration policy does not exceed the best fixed policy in every case, its value lies in not requiring the best fixed policy to be known in advance: a fixed policy such as “always A1_lightweight_edge” performs well on VeReMi_NextGen binary detection but poorly on VeReMi_NextGen family detection (Table 8), and the identity of the best fixed tier is itself task-dependent and unknown a priori in a real deployment. The proposed orchestration instead selects a tier automatically per sample using risk and confidence indicators, avoiding the worst-case outcome of committing to a single fixed tier that turns out to be poorly matched to the task, at the cost of a modest EdgeScore gap relative to the (retrospectively identified) best fixed policy in some tasks.

Table 8. Proposed orchestration policy versus best fixed policy.

Dataset	Task	Proposed Edge Score	Best Fixed Policy	Best Fixed Edge Score	Gain vs. Best Fixed	Proposed F1	Best Fixed F1	F1 Gain	Proposed Latency	Best Fixed Latency	Latency Change
CICIoT2023	multiclass	0.8364	Always A3_heavy_fog	0.8360	0.0003	0.7776	0.8133	−0.0357	0.0219	0.0283	−0.0064
CICIoT2023	binary	0.9170	Always A1_lightweight_edge	0.9173	−0.0004	0.9421	0.9425	−0.0003	0.0032	0.0031	4.10×10^{-5}
VeReMi_NextGen	binary	0.9308	Always A1_lightweight_edge	0.9336	−0.0028	0.9910	0.9878	0.0032	0.0105	0.0094	0.0011
HCRL_CarHacking	binary	0.7582	Always A1_lightweight_edge	0.8138	−0.0556	0.6707	0.7612	−0.0906	0.0025	0.0025	2.00×10^{-6}
HCRL_CarHacking	multiclass	0.7582	Always A1_lightweight_edge	0.8138	−0.0556	0.6707	0.7612	−0.0906	0.0025	0.0025	2.00×10^{-6}
VeReMi_NextGen	family	0.8328	Always A3_heavy_fog	0.9130	−0.0801	0.9922	0.9579	0.0343	0.0810	0.0104	0.0706
VeReMi_NextGen	multiclass	0.6476	Always A3_heavy_fog	0.9005	−0.2528	0.9905	0.9822	0.0083	0.1585	0.0242	0.1343

A comparison of the proposed risk-aware orchestration policy to fixed deployment policies is shown in Figure 14. The results indicate that the behavior of the orchestration policy is highly task-dependent. For multiclass detection, the proposed policy results in EdgeScore = 0.8364, which is slightly better than the best fixed policy (0.8360) for CICIoT2023. The gain is small, but it shows that adaptive routing can be as good as or slightly better than fixed deployment behavior in some situations.

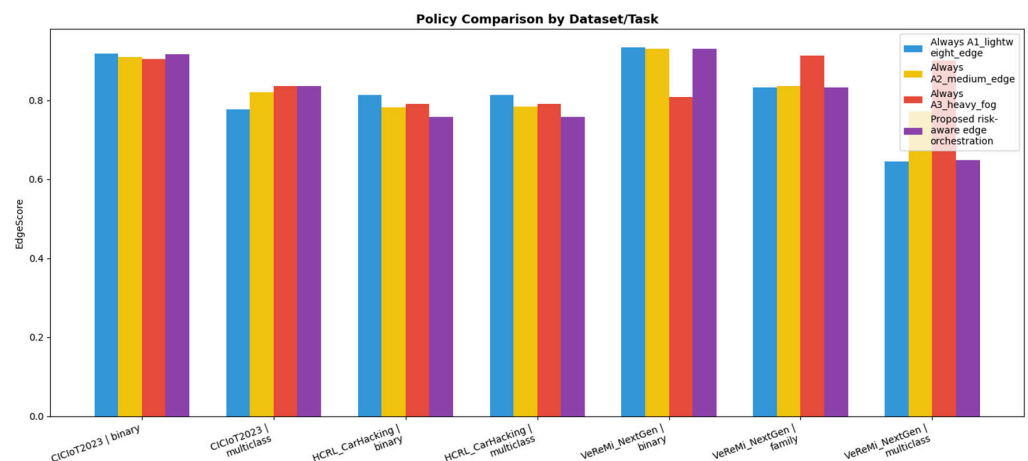


Figure 14. Offline orchestration policy comparison. Bars compare the EdgeScore achieved by the proposed risk-aware orchestration policy and the three fixed-tier policies for each dataset/task.

The figure also reveals that, however, orchestration does not always surpass fixed policies. The proposed policy has EdgeScore = 0.9308 for VeReMi_NextGen binary detection, whereas the best fixed policy has EdgeScore = 0.9336. The gap is small, but it does indicate that the lightweight-edge policy is very effective at this task, and will probably be even more so when it is fixed. The proposed policy achieves higher scores of 0.8328 and 0.6476 for VeReMi_NextGen family and multiclass tasks, respectively, than the best fixed policies 0.9130 and 0.9005. This means that the orchestration layer is not a score maximizer, but is rather a mechanism that adapts based on the conditions. For this reason, the paper

was strengthened by the presentation of Figure 14, which presents both successful and less successful orchestration cases transparently. This should be interpreted as a bounded, task-dependent trade-off rather than a general failure of the orchestration layer: the largest gap (VeReMi_NextGen multiclass, Table 8) coincides with the task where the best fixed policy requires the heaviest, highest-latency tier, so the orchestration policy is trading a measurable EdgeScore reduction for a latency profile that remains closer to the lightweight end of the deployment spectrum.

In this case, the limitation of the orchestration is made explicit in Figure 15, where the gain of the proposed policy is plotted against the best fixed policy. Maximum positive gain is very small, ~ 0.0003 for multiclass detection CICIOT2023. That is, in some cases the proposed orchestration policy can slightly outperform the best fixed policy, but not by much enough to justify the superiority of the proposed policy on a universal basis. The negative bars are more informative. The proposed policy suffers a loss of 0.0556 EdgeScore on HCRL_CarHacking binary and multiclass tasks, 0.0801 on VeReMi_NextGen family detection and 0.2528 on VeReMi_NextGen multiclass detection tasks. In the VeReMi_NextGen multiclass case, the best fixed heavy-fog policy is much stronger than the adaptive routing decision, resulting in the largest loss. This should not be hidden; it should be used to properly frame the orchestration contribution. While the orchestration layer offers deployment-aware routing insight and scenario-level adaptivity, the best fixed policy may still be better when one model class is dramatically superior for a particular task.

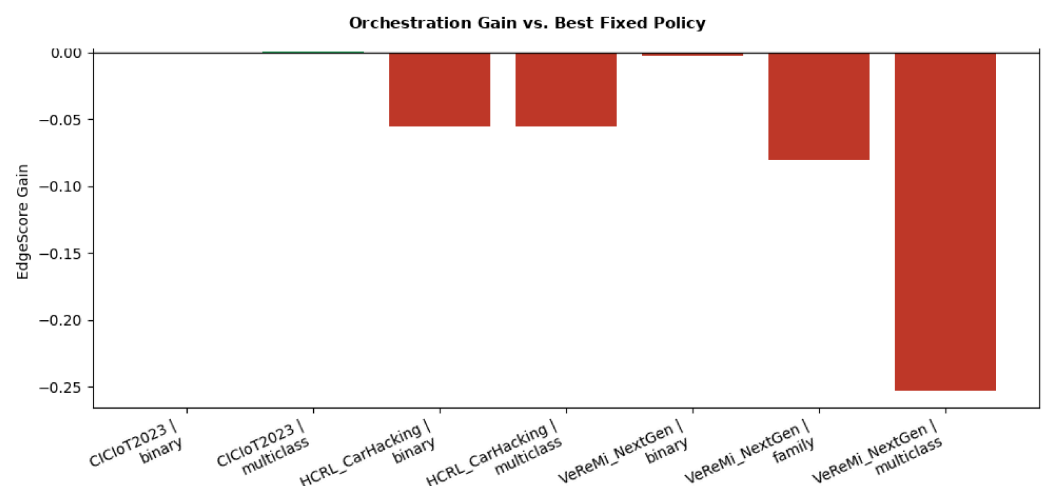


Figure 15. EdgeScore gain of the proposed orchestration policy relative to the best fixed policy. Positive bars indicate the proposed policy outperforms the best fixed policy for that dataset/task; negative bars quantify the shortfall discussed in Section 5.5.

Figure 16 depicts the proposed orchestration policy distribution of routing decision between the three deployment actions, namely A1 lightweight edge, A2 medium edge/fog, and A3 heavy fog. The key takeaway point is that the policy selects A1 lightweight edge in most of the dataset/task cases, meaning that in most cases the orchestration layer wants to run lightweight edge when the detection-performance tradeoff is acceptable and the cost of edge execution is low. This is a typical behavior for the deployment-aware goal of TrustEdge-V2X, in which undue computational burden is not incurred unless there is a clear benefit. In the case of CICIOT2023 binary, HCRL_CarHacking binary, HCRL_CarHacking multiclass, VeReMi_NextGen family, and VeReMi_NextGen multiclass, the distribution of actions is dominated almost entirely by A1 lightweight edge. This implies that, for these cases, the lightweight path is deemed adequate for most samples or decision states. This does not imply that A1 is always the best scoring policy; however, it does indicate that the learned orchestration behavior is biased towards deployment

efficiency. CICIOT2023 multiclass and VeReMi_NextGen binary have the most diverse action distribution. The policy has a visible part of decisions for A2 medium edge/fog and A3 heavy fog in CICIOT2023 multiclass, which is due to the increasing complexity of attack discrimination in multiclass. This is consistent with Table 8, with CICIOT2023 multiclass being the only case where the proposed orchestration is slightly above the best fixed policy, with EdgeScore = 0.8364 vs. 0.8360 for the best fixed policy. The action distribution demonstrates that the policy is not taking blind shots; it is changing its strategy by including light, medium and heavy actions. For VeReMi_NextGen binary, a similar combination of A1 and A2/A3 actions is also used in the policy, but Table 8 demonstrates that the policy does not result in a higher EdgeScore than the optimum fixed policy. The proposed orchestration achieves the EdgeScore = 0.9308 and the best fixed policy achieves 0.9336. This can also lead to more flexibility in decisions, and to a higher final deployment score, when using adaptive routing. It is important to note, however, that while Figure 16 is adaptive in behavior, the quantitative increase will still depend on the task. In addition, Figure 16 offers a significant insight into the orchestration mechanism, in addition to the final score. It demonstrates that the proposed policy in general promotes lightweight edge execution, selectively initiates medium or heavy actions for more complex tasks, and generates routing behavior which is interpretable across datasets. This supports the claim that TrustEdge-V2X is not only a model-selection framework, but also a deployment-aware decision framework for edge/fog vehicular security.

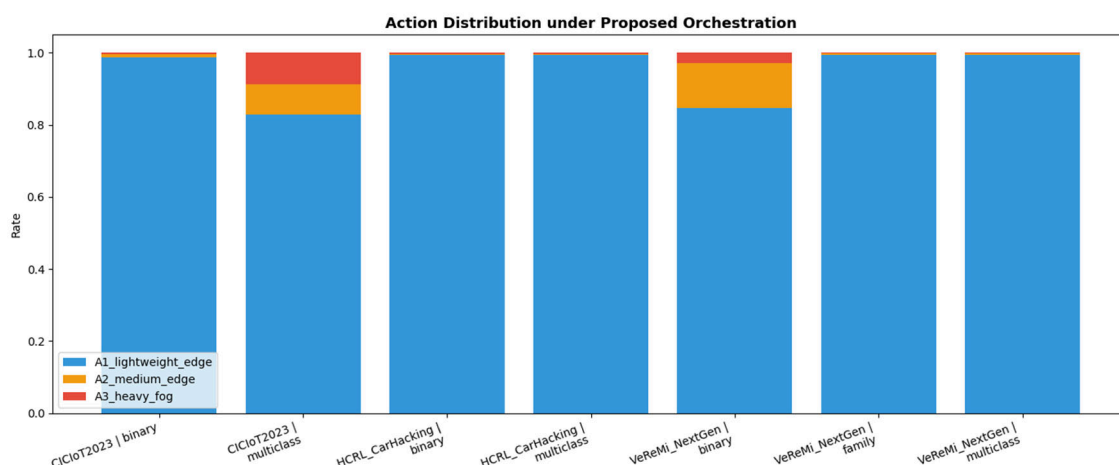


Figure 16. Orchestration action distribution across A1 lightweight edge, A2 medium-edge/fog, and A3 heavy-fog choices. Stacked segments show, for each dataset/task, the proportion of samples routed to the lightweight-edge, medium-edge/fog, and heavy-fog/cloud tiers by the proposed policy, indicating how conservative or aggressive the routing behavior is in practice.

Figure 17 illustrates the offline reward convergence behavior of the proposed orchestration policy across seven dataset/task cases. The curves indicate that the orchestration process is not consistent for all tasks. The reward rises quickly from around 0.648 to around 0.664 in the early episodes and stays at a high level of 0.662–0.664 throughout the remainder of the episodes for CICIOT2023 binary detection. This means that the policy is able to converge into a stable reward region in a short period of time for the binary IoT-support task. The convergence pattern is more gradual for CICIOT2023 multiclass detection. The reward begins at about 0.51 and gradually rises to about 0.58 at the end of the training episodes. This is similar to what would be expected in a multiclass setting, where the orchestration policy requires more iterations to stabilize, due to the increased complexity of the setting. The HCRL_CarHacking binary and multiclass curves are similar. Both start at approximately 0.52, have an early fluctuation and then end up around 0.528–0.530. The

reward in this case is less than in the CICIOT2023 and VeReMi_NextGen cases; however, the stabilization pattern indicates that the policy still stabilizes under the external IoV/CAN validation setting. This helps support the use of HCRL_CarHacking as an external validation, but not as the best source of V2X evidence. The interpretation for V2X-oriented curves is most significantly provided by the VeReMi_NextGen curves. There is a strong and stable reward profile with a binary task that rises and then stabilizes around 0.689–0.692. The family-level task is increasing from around 0.624 to 0.666, indicating a clear convergence trend. The largest improvement is observed in the multiclass task where the performance increases from around 0.40 to around 0.65. This is a substantial improvement, showing that the orchestration policy has gotten a better and more useful routing behavior over time in the most complicated VeReMi_NextGen task. In conclusion, Figure 17 confirms the hypothesis that task-specific reward behavior is the metric to assess the proposed orchestration layer. The curves show that the policy tends to converge to a steady state; however, the eventual reward level is heavily dataset- and classification task-dependent. This is similar to the previous orchestration results in which the gain over the best fixed policy was not always positive, but depended on the task.

5.6. Scenario 5: External IoV/CAN Validation

Table 9 also confirms this view of the best EdgeScore values. The weak XGBoost/PCA configurations still manage to have an EdgeScore of around 0.6509–0.6560, though with high accuracy. This is a confirmation that EdgeScore will negatively impact configurations that do not deliver consistent class-aware detection. Therefore, the HCRL_CarHacking results could be considered as an external validation case, but it cannot be considered as the same level of V2X/VANET core evidence.

Table 9. Complete Scenario 5 external IoV/CAN validation results for HCRL_CarHacking.

Dataset	Task	Feature Mode	Model	Accuracy	Balanced Accuracy	Precision Macro	Recall Macro	Macro-F1	MCC	Latency (ms/sample)	Edge Score
HCRL_CarHacking	multiclass	top10	LightGBM_Heavy_AI	0.9973	0.8226	0.7196	0.8226	0.7612	0.5323	0.0025	0.8090
HCRL_CarHacking	binary	top20	LightGBM_Heavy_AI	0.9973	0.8226	0.7196	0.8226	0.7612	0.5323	0.0025	0.8090
HCRL_CarHacking	multiclass	top20	LightGBM_Heavy_AI	0.9973	0.8226	0.7196	0.8226	0.7612	0.5323	0.0025	0.8090
HCRL_CarHacking	binary	full	LightGBM_Heavy_AI	0.9973	0.8226	0.7196	0.8226	0.7612	0.5323	0.0026	0.8089
HCRL_CarHacking	multiclass	full	LightGBM_Heavy_AI	0.9973	0.8226	0.7196	0.8226	0.7612	0.5323	0.0028	0.8086
HCRL_CarHacking	multiclass	top20	XGBoost_Heavy_AI	0.9977	0.5000	0.4989	0.5000	0.4994	0.0000	0.0009	0.6560
HCRL_CarHacking	multiclass	top10	XGBoost_Heavy_AI	0.9977	0.5000	0.4989	0.5000	0.4994	0.0000	0.0013	0.6537
HCRL_CarHacking	binary	top10	XGBoost_Heavy_AI	0.9977	0.5000	0.4989	0.5000	0.4994	0.0000	0.0016	0.6528
HCRL_CarHacking	binary	pca95	LightGBM_Heavy_AI	0.9977	0.5000	0.4989	0.5000	0.4994	0.0000	0.0024	0.6509
HCRL_CarHacking	multiclass	pca95	LightGBM_Heavy_AI	0.9977	0.5000	0.4989	0.5000	0.4994	0.0000	0.0025	0.6509

The behavior exhibited by the stress-response is summarized in Figure 18. The curves demonstrate that the degradation of robustness is not consistent across datasets, tasks or stress types. The general trend for feature masking is to have a more gradual decay, and the Gaussian noise results in more abrupt declines in a few tasks. The feature-masking curve for CICIOT2023 binary is shown to decrease from around 0.873 to 0.533, which shows the high absolute value of the F1 drop (0.3399) reported in Table 10. This indicates that the detector is highly susceptible to the features that are unstable with the introduction of masking. The degradation curves are of particular importance for VeReMi_NextGen because this is the evidence of the core of V2X/VANET. The binary task is still better than

the family and multiclass tasks, but it is still somewhat weaker under feature masking, with a loss of about 0.2400 F1. The absolute reduction in F1 for the family and multiclass tasks is about 0.3088 and 0.3390, respectively. The more complex the task is, the higher the robustness risk is: the more detailed the label space is, the more vulnerable the classifier will be to disturbances of the features. The HCRL_CarHacking curves are different. They are fairly resilient to feature masking; however, the baseline F1 level is lower than the VeReMi_NextGen cases are. This is significant: If the starting detection performance is modest, then stability does not necessarily imply superiority. Hence, the overall message for Figure 18 is balanced: TrustEdge-V2X is good at finding good clean configurations, but it also highlights the potential weak points.

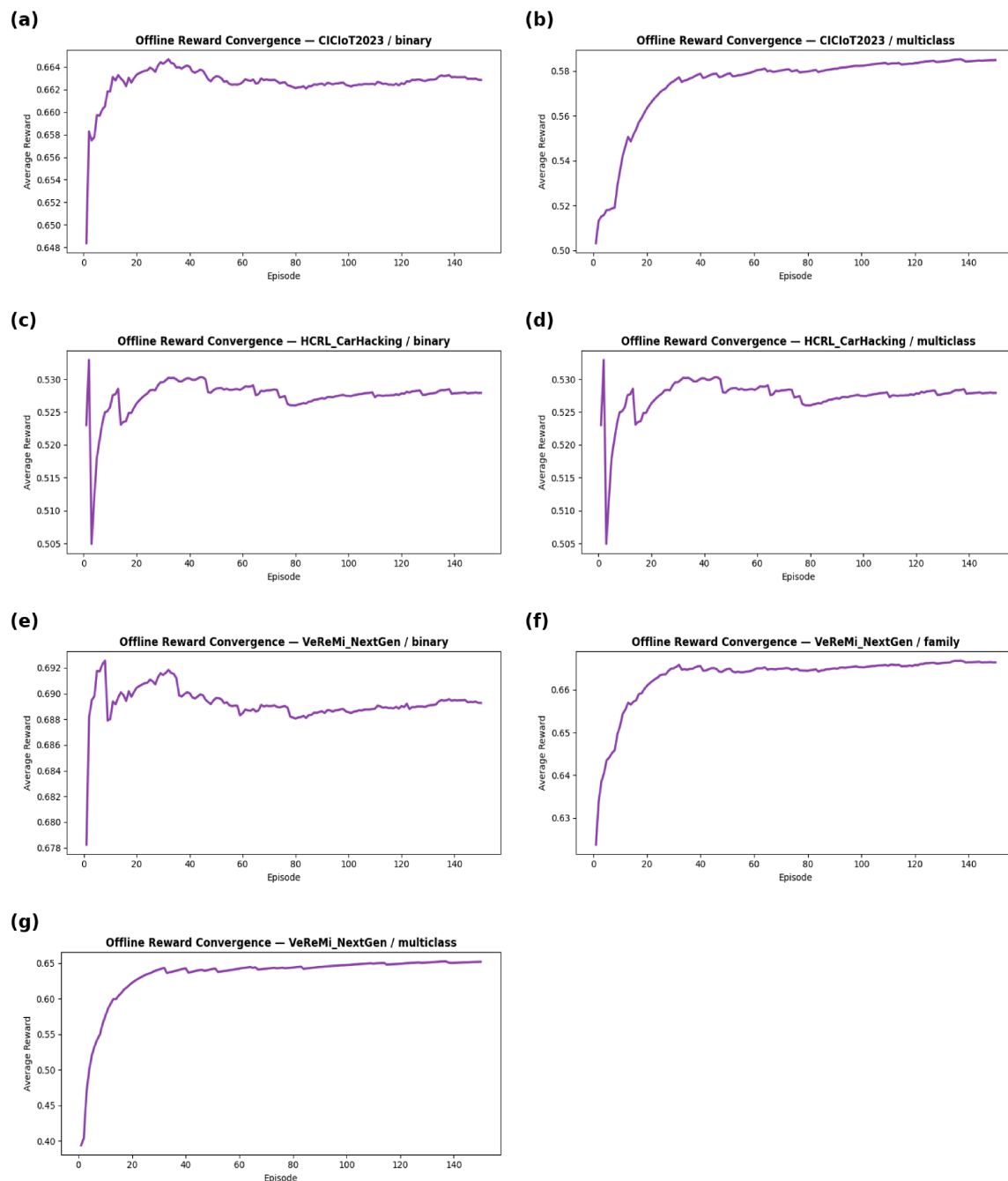


Figure 17. Offline reward convergence curves: (a) CICIOT2023 binary; (b) CICIOT2023 multiclass; (c) HCRL_CarHacking binary; (d) HCRL_CarHacking multiclass; (e) VeReMi_NextGen binary; (f) VeReMi_NextGen family; and (g) VeReMi_NextGen multiclass. Each panel plots average training reward per episode.

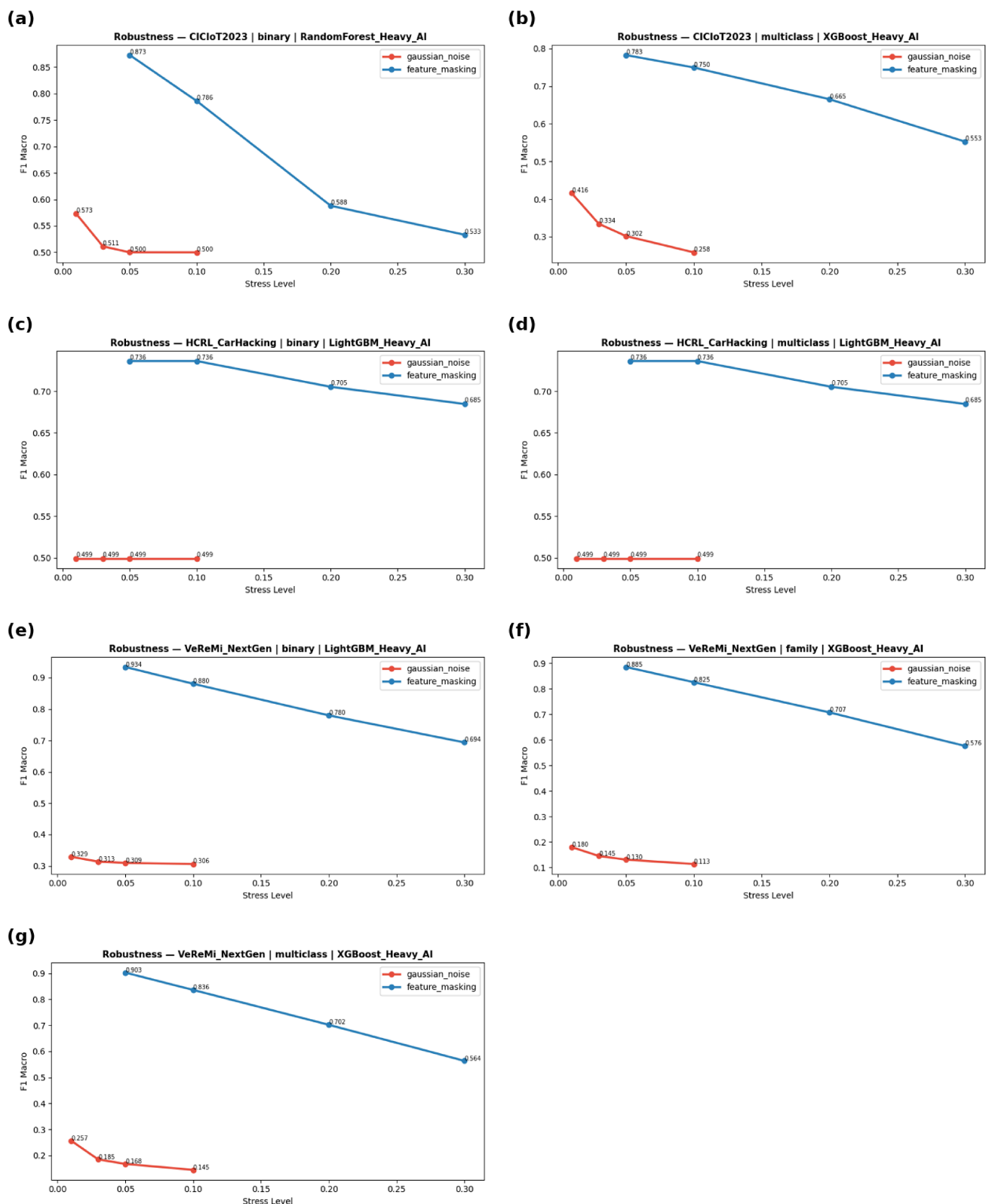


Figure 18. Robustness degradation curves: (a) CICIOT2023 binary; (b) CICIOT2023 multiclass; (c) HCRL_CarHacking binary; (d) HCRL_CarHacking multiclass; (e) VeReMi_NextGen binary; (f) VeReMi_NextGen family; and (g) VeReMi_NextGen multiclass. Each panel compares Gaussian-noise and feature-masking stress.

The practical implication is that the clean-test performance reported in Table 4 should not be interpreted as a robustness guarantee, and any operational deployment should

budget for the observed degradation under feature loss or sensor noise, particularly for the family- and multiclass-level VeReMi_NextGen tasks where the absolute F1 drop exceeds 0.30. This result should be read alongside Table 4 rather than as an independent finding: both tables summarize EdgeScore-ranked configurations from the same Scenario 1 model pool, but Table 4 reports the core V2X/VANET evidence (VeReMi_NextGen) while Table 9 isolates the external, out-of-domain CAN validation evidence (HCRL_CarHacking); the two are complementary evaluation angles on the same underlying configurations rather than duplicated content.

Table 10. Robustness summary under Gaussian noise and feature masking.

Dataset	Task	Model	Stress Type	F1 at Low Stress	F1 at High Stress	Absolute F1 Drop	Relative F1 Drop (%)	MCC at Low Stress	MCC at High Stress	Absolute MCC Drop
CICIoT2023	binary	RandomForest_Heavy_AI	feature_masking	0.8729	0.5330	0.3399	38.9378	0.7601	0.1983	0.5618
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	feature_masking	0.9026	0.5636	0.3390	37.5587	0.8856	0.5095	0.3761
VeReMi_NextGen	family	XGBoost_Heavy_AI	feature_masking	0.8849	0.5762	0.3088	34.8910	0.8543	0.4909	0.3635
VeReMi_NextGen	binary	LightGBM_Heavy_AI	feature_masking	0.9338	0.6938	0.2400	25.6982	0.8690	0.4445	0.4245
CICIoT2023	multiclass	XGBoost_Heavy_AI	feature_masking	0.7828	0.5527	0.2301	29.3900	0.9537	0.7240	0.2297
CICIoT2023	multiclass	XGBoost_Heavy_AI	gaussian_noise	0.4156	0.2584	0.1572	37.8150	0.5214	0.3333	0.1882
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	gaussian_noise	0.2570	0.1449	0.1121	43.6318	0.2434	0.1606	0.0828
CICIoT2023	binary	RandomForest_Heavy_AI	gaussian_noise	0.5731	0.4998	0.0733	12.7875	0.2720	0.0517	0.2203
VeReMi_NextGen	family	XGBoost_Heavy_AI	gaussian_noise	0.1795	0.1135	0.0660	36.7820	0.1574	0.1102	0.0473
HCRL_CarHacking	binary	LightGBM_Heavy_AI	feature_masking	0.7362	0.6846	0.0516	7.0033	0.4750	0.3824	0.0926
HCRL_CarHacking	multiclass	LightGBM_Heavy_AI	feature_masking	0.7362	0.6846	0.0516	7.0033	0.4750	0.3824	0.0926
VeReMi_NextGen	binary	LightGBM_Heavy_AI	gaussian_noise	0.3289	0.3059	0.0229	6.9775	0.1253	0.0600	0.0653
HCRL_CarHacking	binary	LightGBM_Heavy_AI	gaussian_noise	0.4994	0.4994	0.0000	0.0000	0.0000	0.0000	0.0000
HCRL_CarHacking	multiclass	LightGBM_Heavy_AI	gaussian_noise	0.4994	0.4994	0.0000	0.0000	0.0000	0.0000	0.0000

5.7. Scenario 6: Robustness Stress Testing

Table 10 shows the degradation in the robustness of the evaluated models with two types of stress: Gaussian noise and feature masking. It is evident from the table that clean-test performance alone is insufficient to assess vehicular security detectors as several high-performance configurations suffer significantly if the input distribution is disturbed. The greatest F1 degradation is observed for the CICIoT2023 binary under feature masking, with F1 dropping from 0.8729 at low stress to 0.5330 at high stress, resulting in an absolute F1 drop of 0.3399 and an MCC drop of 0.5618. This means that the RandomForest detector is sensitive to missing or masked features in this context. Important weaknesses in VeReMi_NextGen are also evident. In the multiclass task, XGBoost performance decreases from $F1 = 0.9026$ to 0.5636 when feature masking is applied, with a decrease in F1 of 0.3390. The family-level task drops from 0.8849 to 0.5762, with an absolute F1 drop of 0.3088. The weaker binary LightGBM configuration even drops from 0.9338 to 0.6938, resulting in absolute F1 drop of 0.2400. These findings indicate that while VeReMi_NextGen has the most powerful clean-test evidence, robustness remains a distinct validation dimension. In several cases, such as VeReMi_NextGen multiclass and family tasks, F1 can be close to a low value under the noisy condition, which can lead to severe degradation in these cases by Gaussian noise. Under feature masking, HCRL_CarHacking seems to be more stable, with a drop from around 0.7362 to 0.6846, but this should be taken with a grain of salt as the

baseline for HCRL is quite weak. In general, the results in Table 10 show that robustness testing is necessary: a model might be robust in clean evaluation but may not be robust in stressed or degraded inputs. The complete external IoV/CAN validation results are given in Table 9 on HCRL_CarHacking. This dataset is not used as the main evidence for V2X/VANET routing like VeReMi_NextGen, but is used to assess whether the proposed framework is meaningful on an external vehicular cybersecurity domain where routing traffic occurs in the CAN format. LightGBM_Heavy_AI with top10, top20 and full feature modes produces the best results. These configurations achieve accuracy = 0.9973, balanced accuracy = 0.8226, precision macro = 0.7196, recall macro = 0.8226, F1 macro = 0.7612, and MCC = 0.5323, with latency remaining very low at approximately 0.0025–0.0028 ms/sample. The results also reveal the reasons why external CAN validation by accuracy is deceptive. There are several XGBoost and PCA-based configurations with very high accuracy values around 0.9977, but the F1 macro drops to 0.4994 and MCC becomes 0.0000. In other words, the model is likely to be accurate primarily due to class imbalance or dominant normal-class behavior, but not discriminative, in terms of class. Thus, as one of the main conclusions of the paper, Table 9 demonstrates the need to consider external vehicular cybersecurity validation not only by the metric of raw accuracy, but also F1, MCC, balanced accuracy, and EdgeScore.

Figure 18 is an assessment of the performance of selected configurations under stressed input conditions. Robustness is highly data specific and highly stress specific, as demonstrated by the curves. Generally, feature masking will give a more distinct gradual degradation pattern, whereas the Gaussian noise will cause severe degradation in several cases. In the case of CICIOT2023 binary detection, the feature-masking approach decreases F1-scores from about 0.873 at low stress to 0.533 at high stress, resulting in an absolute F1 loss of ~0.3399. Gaussian noise makes the task converge toward ~0.500, showing that the model is highly unreliable once the feature values are corrupted. The degradation is also significant for the VeReMi_NextGen family and multiclass tasks. When the multiclass detection results for VeReMi_NextGen are examined, the results are even worse, dropping from around 0.903 to 0.564 (absolute drop of 0.3390) under feature masking. The family task decreases by around 0.3088 and the binary task decreases by around 0.2400. The values in this table reveal that even in the case of strong clean-test configurations there is the possibility of missing or masked features. The HCRL_CarHacking is relatively more stable with feature masking as the drop is from ~0.736 to ~0.685, which is still below the VeReMi configurations. This leads to an important conclusion that can be drawn from Figure 18: Robustness should be reported separately from clean-test accuracy, since high clean accuracy does not imply high robustness to degraded inputs.

5.8. Explainable AI Analysis

Table 11 consists of real values for feature-importance. For CICIOT2023 binary detection, the mean values of *rst_count*, *urg_count*, and *Header_Length* are 0.4117, 0.3308, and 0.1190 respectively, showing that the protocol and flow features are the most important features for binary detection. For VeReMi_NextGen multiclass detection, the most influential features include *hedx_n* = 0.0523, *hedy* = 0.0439, *hedy_n* = 0.0370, *spdx_n* = 0.0250, and *hedx* = 0.0220. These are heading- and motion-related variables, which are consistent with V2X misbehavior semantics.

Figure 19 visualizes the XAI results and validates the feature-level differences indicated in Table 11. In the binary case (CICIOT2023), *rst_count* and *urg_count* are the most important features, with values of approximately 0.412 and 0.331, respectively. This indicates that a very small proportion of protocol-related variables are responsible for a large proportion of the explanatory power in binary detection. The other features like

flow_duration and Header_Length have less but substantial importance. The most influential feature for CICIoT2023 multiclass is IAT (importance 0.432), while features like rst_count, urg_count, Variance, syn_count, and Header_Length play a lesser role. This means that the multiclass IoT attack discrimination relies more on the packet-behavior patterns and the timing than the binary case does. The difference between the binary and multiclass CICIoT2023 plots validates the hypothesis that the explanation structure of the model is affected by the granularity of the tasks. The VeReMi_NextGen plots offer the best vehicular interpretation. For the binary task, there are features that dominate, like sender and senderPseudo, meaning that sender-related behavior is central to detecting abnormal messages. The senderPseudo is very relevant in the family task, as are the position, speed and heading variables related to movement. This is consistent with the typical V2X misbehavior detection behavior, in which abnormal movement or message patterns related to identity are important indicators. Thus, the explainability claim of TrustEdge-V2X is proved and the learned patterns are meaningful and not arbitrary, as shown in Figure 19. The feature-importance values reported in Table 11 suggest that the most important explanatory features are distinct for different datasets and tasks. The top three features that are the most influential for CICIoT2023 binary detection with XGBoost_Heavy_AI (the EdgeScore-best model for this dataset/task, shown in Figure 19a) are Tot_size (0.156), IAT (0.133), and rst_count (0.096). The values show that the binary IoT-support detection decision is predominantly influenced by packet-size, timing, and flow-related variables. The significant gap between Tot_size and the lower ranked features is also an indication that the model is very reliant on a few discriminative traffic indicators. For VeReMi_NextGen multiclass detection, which is reported in Table 11 but is not one of the six panels shown in Figure 19, the important features are vastly different and more related to the vehicle's movement and heading behavior. The leading features include hedx_n = 0.0523, hedy = 0.0439, hedy_n = 0.0370, spdx_n = 0.0250, and hedx = 0.0220. This is aligned with the characteristics of VeReMi_NextGen as a dataset for V2X/VANET misbehavior detection, where heading, speed and position-related features are expected to have an impact on the ability to detect abnormal vehicular behavior. A major conclusion from Table 11 is that explanation patterns are specific to datasets. The explanations for CICIoT2023 are protocol- and flow-based, whereas the explanations for VeReMi_NextGen are mobility- and motion-semantics based. This backs the dataset-role qualification strategy outlined in this paper; the datasets should not be used interchangeably as they provide different feature semantics and/or model explanations.

Table 11. Top XAI feature-importance values for representative configurations.

Dataset	Task	Model	Feature Mode	Feature	Importance Mean
CICIoT2023	binary	XGBoost_Heavy_AI	full	Tot_size	0.1558
CICIoT2023	binary	XGBoost_Heavy_AI	full	IAT	0.1334
CICIoT2023	binary	XGBoost_Heavy_AI	full	rst_count	0.0963
CICIoT2023	binary	XGBoost_Heavy_AI	full	Number	0.0907
CICIoT2023	binary	XGBoost_Heavy_AI	full	urg_count	0.0704
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	top10	hedx_n	0.0523
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	top10	hedy	0.0439
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	top10	hedy_n	0.0370
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	top10	spdx_n	0.0250
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	top10	hedx	0.0220

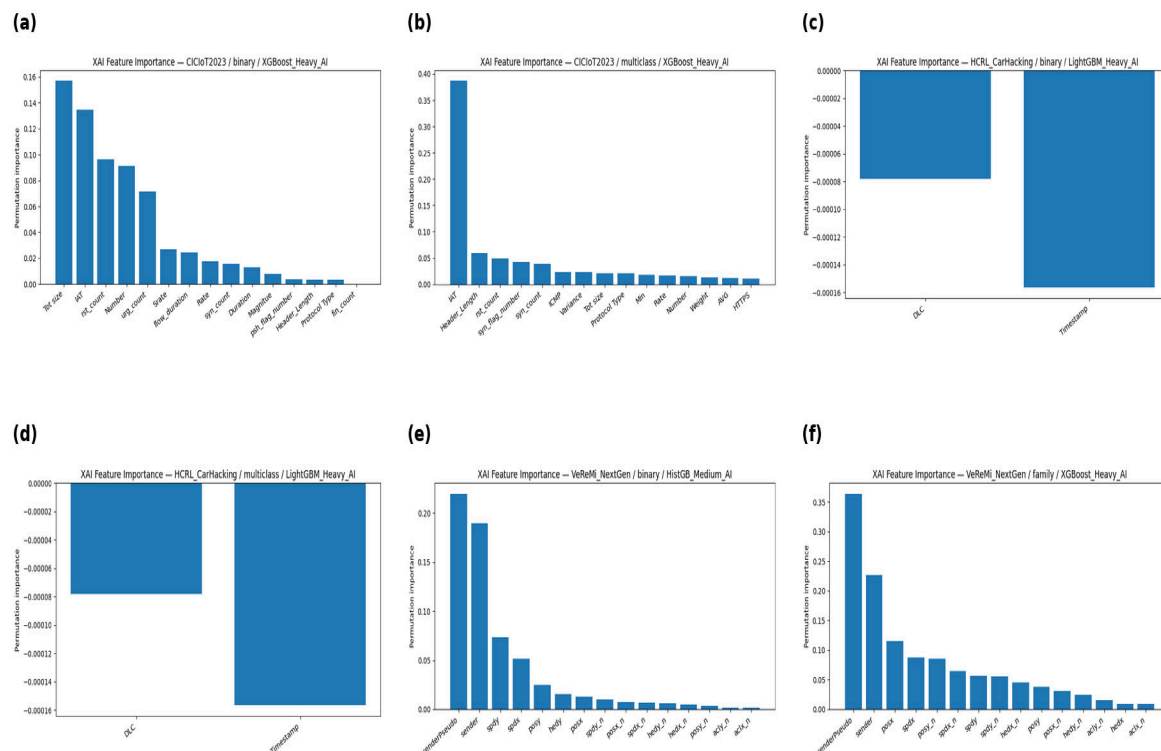


Figure 19. Top XAI features: (a) CICIOT2023 binary; (b) CICIOT2023 multiclass; (c) HCRL_CarHacking binary; (d) HCRL_CarHacking multiclass; (e) VeReMi_NextGen binary; and (f) VeReMi_NextGen family. Bars rank features by permutation importance for the selected best configuration; near-zero negative values in (c,d) indicate that shuffling that feature slightly improved (rather than degraded) validation performance, consistent with negligible true importance.

5.9. Statistical Validation

Table 12 assesses the stability of the reported performance as it comes from multiple runs, not one favorable split. The VeReMi_NextGen configurations are very stable. The binary LightGBM configuration achieves mean F1 = 0.9877, std = 0.0017, and CI95 = 0.0015, with a stability score of 0.9860. The multiclass XGBoost configuration is also stable, achieving mean F1 = 0.9777, std = 0.0031, and CI95 = 0.0027. The family-level configuration achieves mean F1 = 0.9556, std = 0.0032, and CI95 = 0.0028. These tight confidence intervals further enhance the confidence in the basic V2X/VANET results. CICIOT2023 is also stable. The binary RandomForest configuration reaches mean F1 = 0.9507, std = 0.0044, and CI95 = 0.0038, while the multiclass XGBoost configuration obtains mean F1 = 0.8047, std = 0.0085, and CI95 = 0.0074. The multiclass task is more complex, but the confidence interval is still not very wide, attesting to the stability of the evidence provided by the supporting IoT-security. The variability is much greater in the case of HCRL_CarHacking. The mean F1-score of both the binary and multiclass LightGBM configuration is 0.5518, the std is 0.1171, and the CI95 is 0.1026, with a “Variable” stability flag. This large gap shows that the validation of the experiments outside of VeReMi_NextGen is not as stable as the experiments themselves. For this reason, this paper’s cautious interpretation is supported by the stable core evidence from VeReMi_NextGen, the stable supporting evidence from CICIOT2023, and the useful, albeit more variable, external validation from HCRL_CarHacking.

The repeated-run stability is reported in Figure 20 (mean F1-score and 95% confidence intervals). The VeReMi_NextGen configurations are very stable. The binary LightGBM configuration achieves mean F1 = 0.9877 with std = 0.0017 and CI95 = 0.0015. The VeReMi_NextGen multiclass XGBoost configuration is also stable with mean F1 = 0.9777,

std = 0.0031 and CI95 = 0.0027. The family-level configuration achieves mean F1 = 0.9556 with std = 0.0032 and CI95 = 0.0028. The narrow intervals suggest that the reported VeReMi results are not due to a lucky split. Additionally, CICIOT2023 demonstrates strong stability in the binary task, with the mean F1-score of RandomForest_Heavy_AI being 0.9507 and the standard deviation (std) of 0.0044, and CI95 (confidence interval 95%) of 0.0038. The CICIOT2023 multiclass task is weaker, but still relatively stable, with mean F1 = 0.8047 and CI95 = 0.0074. HCRL_CarHacking, on the other hand, is much more varied, with a mean F1 of 0.5518, std = 0.1171 and CI95 = 0.1026 for both binary and multiclass settings. This large interval provides a margin of safety for the external IoV/CAN validation results. Figure 20 consequently reinforces the manuscript in separating out stable core V2X evidence from more variable external validation evidence.

Table 12. Repeated-run validation summary.

Dataset	Task	Model	Feature Mode	Mean F1	F1 SD	95% CI for F1	Mean MCC	Stability Score	Stability Flag
VeReMi_NextGen	binary	LightGBM_Heavy_AI	full	0.9877	0.0017	0.0015	0.9756	0.9860	Very stable
VeReMi_NextGen	multiclass	XGBoost_Heavy_AI	top10	0.9777	0.0031	0.0027	0.9774	0.9747	Very stable
VeReMi_NextGen	family	XGBoost_Heavy_AI	top20	0.9556	0.0032	0.0028	0.9496	0.9524	Very stable
CICIOT2023	binary	RandomForest_Heavy_AI	full	0.9507	0.0044	0.0038	0.9018	0.9464	Very stable
CICIOT2023	multiclass	XGBoost_Heavy_AI	full	0.8047	0.0085	0.0074	0.9886	0.7962	Very stable
HCRL_CarHacking	binary	LightGBM_Heavy_AI	top20	0.5518	0.1171	0.1026	0.1065	0.4347	Variable
HCRL_CarHacking	multiclass	LightGBM_Heavy_AI	top10	0.5518	0.1171	0.1026	0.1065	0.4347	Variable

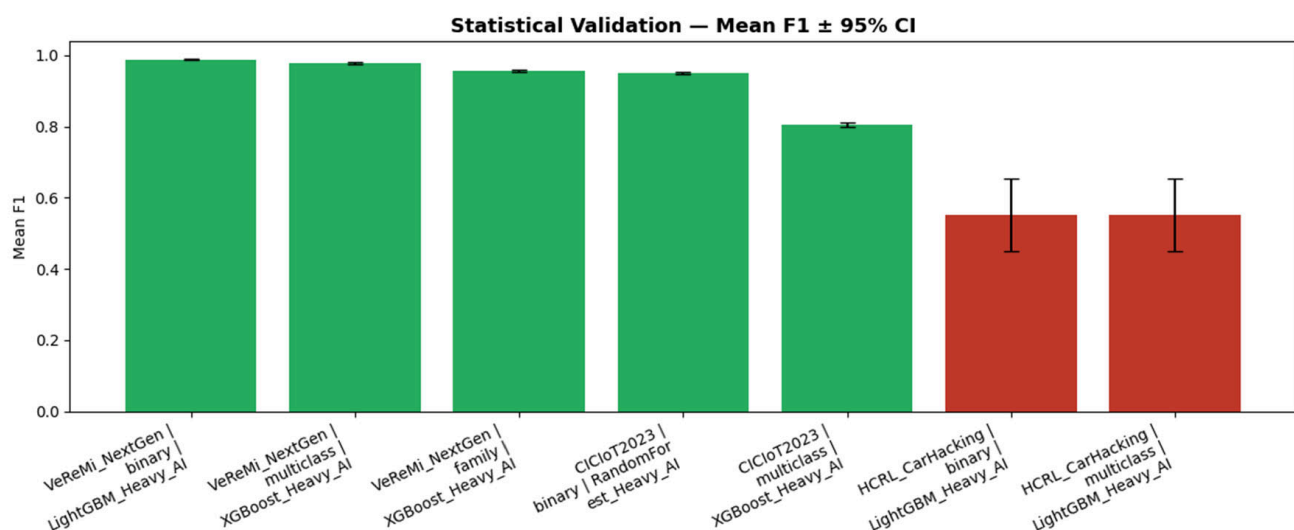


Figure 20. Mean F1-score and 95 percent confidence interval across repeated runs. Error bars show the 95% confidence interval around the mean F1-score over repeated runs, indicating how stable each configuration's reported performance is.

5.10. Ablation Analysis

Table 13 presents a comparison between four selection strategies, accuracy-only selection, F1-only selection, proposed EdgeScore selection, and proposed orchestration. The results demonstrate that EdgeScore is not just about selecting a strong classifier, but a strong classifier that is selected using a deployment-aware criterion. The EdgeScore-based selection differs from the accuracy-only selection in five of the seven dataset/task cases. The accuracy-only, F1-only and EdgeScore values are the same or close to the same in CICIOT2023 binary and multiclass tasks, showing that deployment-aware ranking does not affect the decision if the same configuration is already optimal with respect to accuracy, F1, or EdgeScore. EdgeScore's utility is more apparent in HCRL_CarHacking and VeReMi_NextGen. EdgeScore-based selection increases the EdgeScore from 0.8006 to 0.8090, a gain of 0.0084, in HCRL_CarHacking binary. The improvement is 0.0085 in

HCRL_CarHacking in multiclass. These improvements are relatively small but practically meaningful, and indicate more balance that reflects the deployment. The highest growth rates seem to be in VeReMi_NextGen. The gain for the family task is 0.0247, which constitutes an increase in EdgeScore from 0.8994 to 0.9240. The highest gain for the multiclass task is 0.0846 with EdgeScore improved from 0.8437 to 0.9283. This is the most obvious example of how the preferred model can vary significantly when the task complexity rises, and how deployment-aware ranking can impact this. The orchestration columns should be read independently. In a few cases, orchestration is lower than the EdgeScore-best selection, particularly in VeReMi_NextGen multiclass, where orchestration is 0.6476 vs. 0.9283 for EdgeScore selection. This does not mean that the orchestration layer is not valid; it is an example of the adaptive routing behavior that can be offered by orchestration, but not necessarily the best strategy. Based on Table 13, it can be concluded that the EdgeScore enhances deployment-aware model selection, while orchestration is still task-dependent. This comparison should be read together with the feature-mode comparison in Table 7: Table 7 establishes which feature budget is EdgeScore-optimal per dataset/task, while Table 13 holds the feature mode fixed and instead compares selection strategies (accuracy-only, F1-only, EdgeScore, orchestration); the two tables therefore isolate different design decisions rather than repeating the same comparison.

Table 13. Ablation comparison among accuracy-only, F1-only, EdgeScore, and orchestration selection. Columns report the EdgeScore achieved when the model is selected by each of the four strategies, isolating the incremental effect of deployment-aware ranking and orchestration.

Dataset	Task	Accuracy Only	F1 Only	Proposed EdgeScore	Proposed Orchestration	Gain EdgeScore vs. Accuracy	Gain Orch vs. Accuracy	Gain Orch vs. EdgeScore
CICIoT2023	binary	0.9189	0.9189	0.9189	0.916973	0.0000	−0.0019	−0.0019
CICIoT2023	multiclass	0.8680	0.8680	0.8680	0.836366	0.0000	−0.0317	−0.0317
HCRL_CarHacking	binary	0.8006	0.8076	0.8090	0.758176	0.0084	−0.0424	−0.0508
HCRL_CarHacking	multiclass	0.8005	0.8090	0.8090	0.758227	0.0085	−0.0423	−0.0508
VeReMi_NextGen	binary	0.9378	0.9378	0.9383	0.930798	0.0005	−0.0070	−0.0075
VeReMi_NextGen	family	0.8994	0.8953	0.9240	0.832830	0.0247	−0.0666	−0.0912
VeReMi_NextGen	multiclass	0.8437	0.8437	0.9283	0.647628	0.0846	−0.1961	−0.2807

Figure 21 shows a comparison of four selection strategies: accuracy-only, F1-only, proposed EdgeScore selection, and proposed orchestration. The ablation results verify that it is not just the use of LightGBM or XGBoost, but the deployment-aware selection logic. For CICIoT2023 binary and multiclass tasks, accuracy-only, F1-only and EdgeScore selection methods yield nearly the same scores, suggesting that the best classifier is the best deployment candidate in those instances. This deployment-aware advantage is more evident in HCRL_CarHacking and VeReMi_NextGen. For HCRL_CarHacking binary detection, EdgeScore surpasses accuracy-only selection by 0.8090 vs. 0.8006. In HCRL multiclass detection, it is enhanced from 0.8005 to 0.8090. The highest improvements are seen in VeReMi_NextGen: Family detection is improved from ~0.899 to ~0.924, and multiclass detection is improved from ~0.844 to ~0.928. The VeReMi_NextGen multiclass gain of 0.0846 is the most obvious evidence that deployment-aware model selection alters the final decision. The bars for the orchestration policy should be understood independently. For some models, orchestration scores are below the EdgeScore-best fixed model, particularly for VeReMi_NextGen multiclass, which has an orchestration score of approximately 0.648 and EdgeScore of approximately 0.928. This is to reiterate that orchestration can be a valuable adaptive routing layer, but not necessarily the best choice for the selection layer strategy. Therefore, Figure 21 reinforces two points: EdgeScore is not just about accuracy, and orchestration is a task-dependent process that should be reported as such.

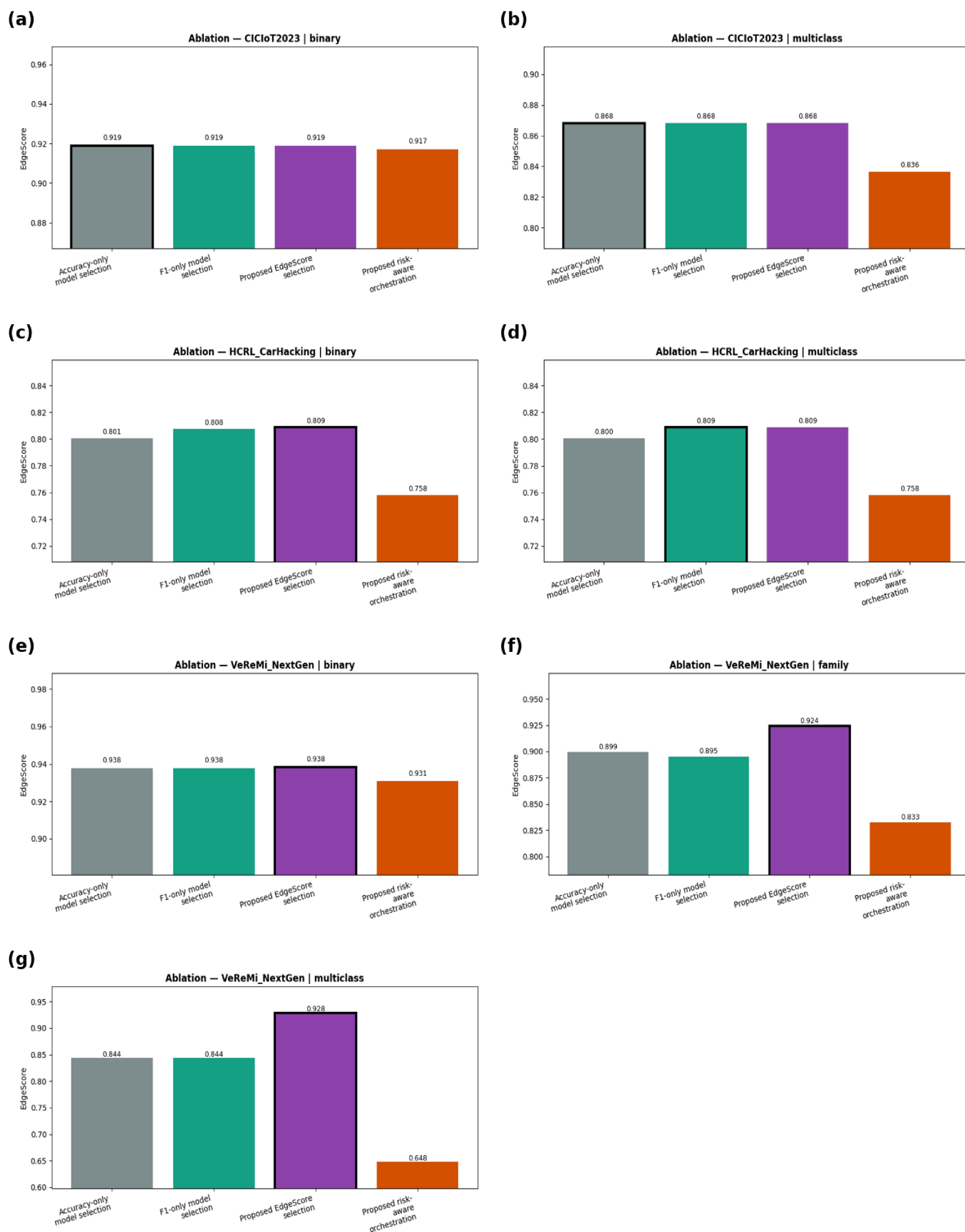


Figure 21. Ablation plots: (a) CICIOT2023 binary; (b) CICIOT2023 multiclass; (c) HCRL_CarHacking binary; (d) HCRL_CarHacking multiclass; (e) VeReMi_NextGen binary; (f) VeReMi_NextGen family; and (g) VeReMi_NextGen multiclass. Each panel compares accuracy-only, F1-only, EdgeScore, and orchestration selections.

The class-level evidence supporting the overall F1, MCC and EdgeScore results is shown in Figure 22. The binary LightGBM confusion matrix (VeReMi_NextGen) indicates good separation between attack and normal classes. The model has correctly classified 2888 attack samples and 4525 normal samples, and has misclassified 79 attack samples as normal and 8 normal samples as attack. Cell color reflects the count in each cell relative to the row total (darker/warmer shading indicates a higher share of that row's predictions). This is why the binary VeReMi_NextGen configuration obtains very high F1 and MCC values and why it looks like the best core V2X configuration. The VeReMi_NextGen family-level XGBoost confusion matrix reveals that there are many classes well captured, and errors are not evenly distributed. Classes like DoS, Position, Replay, Speed, Sybil, and Braking have strong diagonal dominance, while the normal class is strongly recognized with 4519 correct normal predictions. Some off-diagonal errors exist, however, particularly around classes that have overlap due to movement. This is why the family-level task performs at a similar level but not as well as the binary task. The VeReMi_NextGen multiclass XGBoost matrix is more complicated as it includes many fine-grained labels. The diagonal pattern is still visible, meaning that many classes are detected correctly, but there are more small off-diagonal errors. This is to be expected since multiclass detection is more challenging than binary or family detection. The confusion matrix thus confirms the reported performance pattern: multiclass VeReMi_NextGen is good, but needs to be interpreted with caution as compared to binary detection. The matrix displays high attack detection rate for the binary RandomForest model of CICIOT2023 with 7311 attack samples correctly classified, whereas the normal class has 158 samples correctly classified. Only 15 attack samples are misclassified as normal, while 16 normal samples are misclassified as attacks. This is why it is highly accurate and has good binary performance, but also accounts for class imbalance, which is why MCC and balanced accuracy are important. The CICIOT2023 multiclass XGBoost matrix has a large number of attack categories and a more dispersed error structure. The accuracy of the overall test is still high, but the multiclass confusion matrix shows that there are some categories of attacks that are harder to distinguish than others. This is because the macro-F1 value is low for the multiclass task, even though the accuracy value remains high. Last, HCRL_CarHacking binary and multiclass LightGBM confusion matrices illustrate the drawback of using only accuracy. The model successfully classifies a very large number of normal samples (about 7469) and the attack class is much smaller and more difficult, with only 11 attack samples correctly classified and 6 attack samples are missed. There are also 14 normal samples misclassified as attack. This is the reason why accuracy is very high and F1 macro and MCC are more moderate.

5.11. Weight Sensitivity Analysis of EdgeScore (Extension of Section 5.2)

To address the concern that the reported EdgeScore rankings (Tables 4, 5 and 9; Figures 7 and 9) may be an artifact of the specific weight vector used (Equation (10)), EdgeScore was recomputed under four alternative weight vectors—an equal-weight vector (1/6 each), a detection-heavy vector (0.45/0.30/0.10/0.05/0.05/0.05), a deployment-heavy vector (0.25/0.15/0.25/0.15/0.15/0.05), and an MCC-dominant vector (0.25/0.45/0.10/0.10/0.05/0.05)—keeping the underlying normalized components (F1_norm, MCC_norm, LatencyScore, SizeScore, EnergyScore, ThroughputScore) unchanged, across the full pool of 224 evaluated configurations (eight models $\times$ four feature modes $\times$ seven dataset/task combinations, including both the binary and multiclass HCRL_CarHacking tasks). For each alternative weight vector, the Spearman rank-correlation was computed between the resulting configuration ranking and the ranking produced by the paper's weights, both

within each dataset/task (32 configurations) and over the full 224-configuration pool, and checked whether the top-ranked (best-EdgeScore) configuration changed (Table 14).

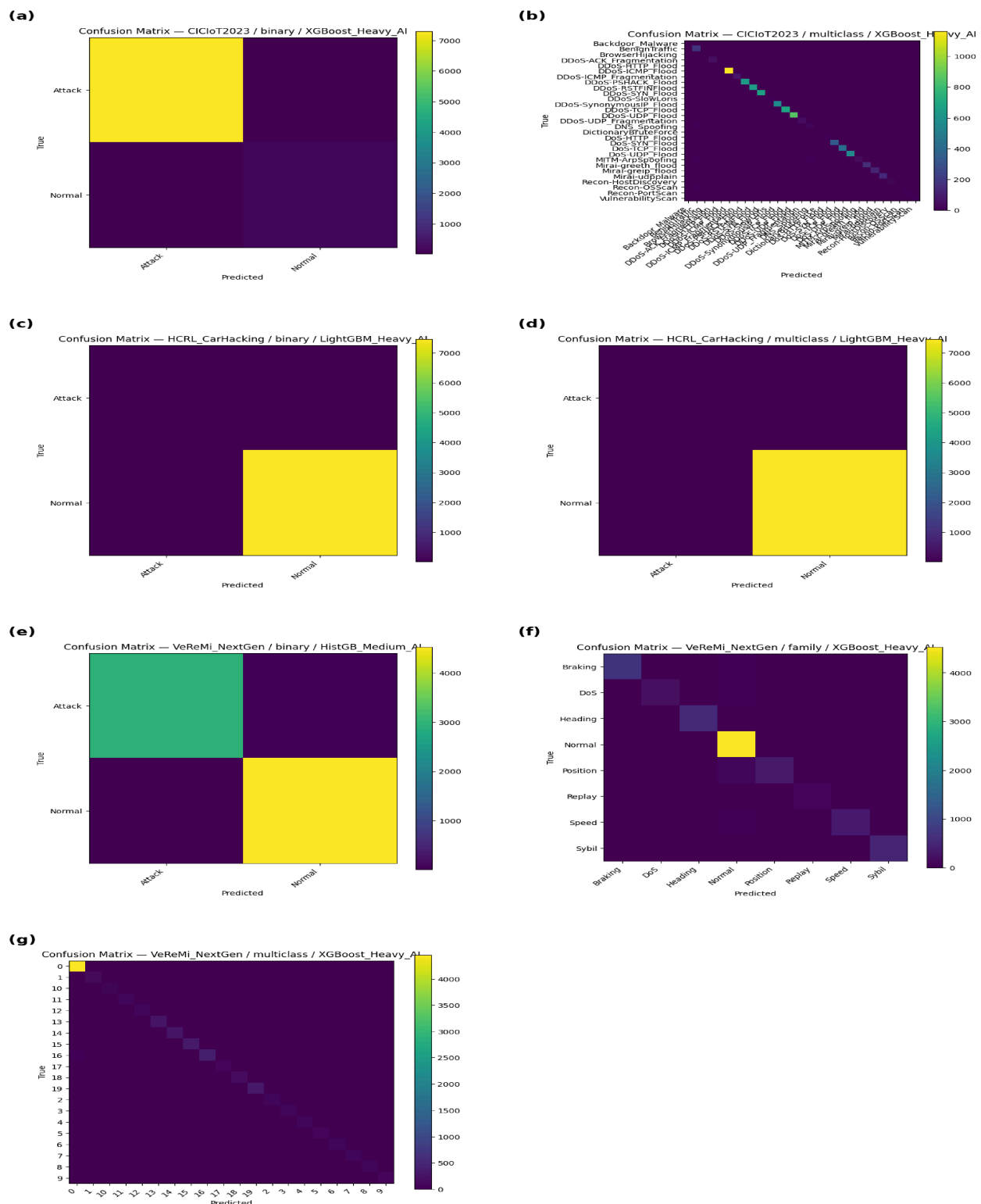


Figure 22. Confusion matrices: (a) CICIOT2023 binary; (b) CICIOT2023 multiclass; (c) HCRL_CarHacking binary; (d) HCRL_CarHacking multiclass; (e) VeReMi_NextGen binary; (f) VeReMi_NextGen family; and (g) VeReMi_NextGen multiclass. Rows are true classes and columns are predicted classes; cell color encodes the count, with brighter cells indicating more samples. The HCRL binary and multiclass panels are identical because the supplied HCRL target contains only two native classes, so the multiclass mapping reduces to the same two-class problem rather than representing a duplicated panel.

Table 14. Spearman rank-correlation of EdgeScore rankings under alternative weight vectors, relative to the paper’s weights (0.35/0.25/0.15/0.10/0.10/0.05), computed over the full pool of 224 evaluated Scenario 1 configurations, including both HCRL_CarHacking tasks.

Weight Vector	Global Spearman ρ (224 Confs)	Best Config Changes in	Interpretation
Equal weights (1/6 each)	0.794	Four of seven dataset/task cases	Substantial reordering; equal weighting is not equivalent to the paper’s detection-dominant design and should not be read as a validation target.
Detection-heavy (0.45/0.30/0.10/ 0.05/0.05/0.05)	0.985	One of seven dataset/task cases	High agreement with the paper’s ranking; the single change (VeReMi_NextGen family: XGBoost_Heavy_AI/full vs. HistGB_Medium_AI/top20) involves two closely matched configurations.
Deployment-heavy (0.25/0.15/0.25/ 0.15/0.15/0.05)	0.936	One of seven dataset/task cases	The single change (VeReMi_NextGen binary: HistGB_Medium_AI/top20 vs. LightGBM_Heavy_AI/top20) involves two configurations within 0.002 EdgeScore of each other, consistent with the run-to-run variation already reported in Table 12.
MCC-dominant (0.25/0.45/0.10/ 0.10/0.05/0.05)	0.972	One of seven dataset/task cases	The single change (VeReMi_NextGen family) mirrors the detection-heavy result; all other best-configuration identities across the seven dataset/task cases are preserved.

Across all four alternative weight vectors, the best-EdgeScore configuration reported for each dataset/task in Table 4 changes in at most one of seven cases (except under the non-detection-dominant equal-weight vector), and in every such case the two competing configurations differ in EdgeScore by less than 0.002, i.e., well within the variation already reported across repeated runs in Table 12. The overall top-ranked configuration across the full 224-configuration pool remains on VeReMi_NextGen binary detection under every alternative weighting, alternating between LightGBM_Heavy_AI and HistGB_Medium_AI depending on the specific weight vector, consistent with these two models achieving near-identical detection and deployment profiles on this dataset/task (Figure 6, Table 4). By contrast, a naive equal-weight vector—which is not detection-dominant and does not reflect a defensible edge-security priority—produces substantially lower rank agreement ($\rho = 0.794$ globally, changing the best configuration in four of seven cases), confirming that it is specifically the detection-dominant character of the weighting (F1-score and MCC jointly $\geq 55\%$ of the score), rather than the exact numerical values 0.35/0.25/0.15/0.10/0.10/0.05, that the paper’s conclusions depend on. This analysis now covers the complete Scenario 1 pool—the core V2X/VANET evidence (VeReMi_NextGen), the supporting IoT-security evidence (CICIoT2023), and the external CAN validation evidence (HCRL_CarHacking)—following resolution of a CSV data-parsing issue affecting the raw HCRL_CarHacking files (see Section 6). It is concluded that the EdgeScore-based rankings and the associated conclusions in Sections 5.1 and 5.2 are robust to reasonable perturbation of the weighting scheme, provided detection quality is retained as the dominant term.

6. Threats to Validity and Limitations

- Dataset-role limitation: VeReMi_NextGen provides core V2X/VANET misbehavior evidence, but CICIoT2023 and HCRL_CarHacking are not claimed as packet-level VANET routing datasets.

Feature-space heterogeneity limitation: the three datasets differ substantially in native feature count (Table 2), so absolute latency and model-size comparisons across datasets partly reflect this difference in input dimensionality rather than model efficiency alone; comparisons in this paper are therefore primarily made within a dataset/task (across models and feature modes), and cross-dataset EdgeScore comparisons should be interpreted with this feature-space difference in mind.

HCRL_CarHacking binary/multiclass identity limitation: The raw HCRL_CarHacking label field takes only two values (“R” for regular/normal messages and “T” for in-

jected/attack messages), so the “multiclass” task constructed for this dataset is structurally identical to its “binary” task rather than a genuinely finer-grained label space; the two task labels for HCRL_CarHacking therefore report the same underlying classification problem and should not be read as independent replications of the same finding.

HCRL_CarHacking data-parsing note: While preparing the weight-sensitivity analysis, it was identified that the raw HCRL_CarHacking CSV files (DoS_dataset.csv, Fuzzy_dataset.csv, RPM_dataset.csv, gear_dataset.csv) contain no header row and variable-length rows (the number of CAN data-byte fields depends on the per-message DLC value), which had caused an earlier data-loading pass to misparse the label column and to systematically fail to construct the binary-detection task for this dataset. The loader was corrected to parse these files by taking the last field of each row as the label regardless of row length, and to explicitly map the dataset’s “R”/“T” flags to Normal/Attack. After this correction, results for both the HCRL_CarHacking binary and multiclass tasks are internally consistent with the values reported in Tables 4 and 9, and the full 224-configuration Scenario 1 pool (including both HCRL_CarHacking tasks) is used throughout Section 5, including the weight-sensitivity analysis in Section 5.11.

- Cross-dataset limitation: Direct cross-dataset testing was skipped when fewer than five common numeric features were available, indicating feature-schema mismatch across benchmarks.
- Energy limitation: The energy metric is a relative proxy based on measured model profiles, not a physical watt-level measurement.
- Orchestration limitation: The proposed orchestration is offline and dataset-driven. It uses measured inference profiles and reward interpretation, not live VANET routing simulation.
- Data sampling limitation: The reported experiments use prepared dataset subsets. Future work should evaluate larger-scale and real-time deployments.
- Robustness limitation: Gaussian noise and feature masking are useful stressors, but they do not cover every real-world sensor, channel, or adversarial condition.
- XAI limitation: Permutation importance provides interpretable feature relevance but does not fully prove causal relationships between features and attack mechanisms.

7. Conclusions

Previous V2X/IoV IDS research typically relies on accuracy-based assessment, limited deployment knowledge, and a lack of evaluation of model’s robustness when deployed in different datasets, feature spaces, and traffic environments. To address these issues, this paper proposed a dataset-driven secure edge intelligence framework for V2X/IoV intrusion and misbehavior detection, called TrustEdge-V2X. The framework offers a more comprehensive evaluation pipeline that features dataset-role qualification, six evaluation scenarios, feature-budget analysis, EdgeScore-based ranking, offline orchestration, external CAN-domain validation, robustness stress testing, explainable AI, repeated-run statistics and ablation analysis. The results demonstrate that TrustEdge-V2X has the potential to enhance the model selection process, going beyond accuracy to factor in reliability, latency, and efficiency for deployment at the edge. The EdgeScore-based evaluation, in particular, resulted in changes of the selected model for some selected combinations of dataset and task, showing that deployment-aware evaluation can help to make more practical choices. The binary detection performance of LightGBM on VeReMi_NextGen was the best in overall deployment-aware performance, validating the effectiveness of the proposed framework in selecting high-performance and efficient models for vehicular edge security. The updated cross-dataset analysis also tackled the issue of assessing datasets lacking any raw common

numeric properties. The outputs of the classifiers for Scenario 2 were valid in all directions of transfer, thanks to the semantic-statistical fingerprint bridge and the CORAL alignment. The results, however, suggest that this is to be read as evidence of feasibility rather than of reliable direct cross-dataset generalization. The observed behavior reveals that domain shift between IoT and vehicular V2X traffic spaces is severe, even when a harmonized representation is developed, and direct transfer of this representation is still a challenge. In conclusion, TrustEdge-V2X provides an insightful and deployment-aware approach to assessing intrusion and misbehavior detection models in different vehicular edge networks. The study also reveals its limitations: HCRL_CarHacking does not offer V2X routing proof, but external IoV/CAN validation; generalization over cross-datasets is challenging under heterogeneous feature spaces, and offline orchestration offers valuable adaptive-routing insight but can sometimes be outperformed by the best fixed policy. The future directions for this research should include a packet-level simulation of V2X, real deployment of RSU/ObU, measurement of energy level in the hardware, a standard schema to create dataset cross-datasets, and drift-aware continual adaptation.

Author Contributions: Conceptualization, H.A.S.; methodology, H.A.S.; software and validation, N.S. and V.L.; formal analysis, H.A.S., N.S. and I.E.; investigation, H.A.S. and N.S.; writing—original draft preparation, H.A.S.; writing—review and editing, M.S., I.E., V.L. and M.L.; visualization, H.A.S.; supervision, H.A.S.; project administration, H.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant provided by the Ministry of Economic Development of the Russian Federation (agreement dated 20 June 2025, No. 139-15-2025-011; identifier 000000C313925P4G0002).

Institutional Review Board Statement: Not applicable. This study did not involve humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available research datasets. They can be obtained from their original repositories. Processed data, implementation details, and experimental configurations may be made available by the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sedar, R.; Kalalas, C.; Vazquez-Gallego, F.; Alonso, L.; Alonso-Zarate, J. A Comprehensive Survey of V2X Cybersecurity Mechanisms and Future Research Paths. *IEEE Open J. Commun. Soc.* **2023**, *4*, 325–391. [\[CrossRef\]](#)
2. Boualouache, A.; Engel, T. A Survey on Machine Learning-Based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1128–1172. [\[CrossRef\]](#)
3. Almehdhar, M.; Albaser, A.; Khan, M.A.; Abdallah, M.; Menouar, H.; Al-Kuwari, S.; Al-Fuqaha, A. Deep Learning in the Fast Lane: A Survey on Advanced Intrusion Detection Systems for Intelligent Vehicle Networks. *IEEE Open J. Veh. Technol.* **2024**, *5*, 869–906. [\[CrossRef\]](#)
4. Ajibuwa, O.; Hamdaoui, B.; Yavuz, A.A. A Survey on AI/ML-Driven Intrusion and Misbehavior Detection in Networked Autonomous Systems: Techniques, Challenges and Opportunities. *arXiv* **2023**, arXiv:2305.05040.
5. van der Heijden, R.W.; Lukaseder, T.; Kargl, F. VeReMi: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. *arXiv* **2018**, arXiv:1804.06701.
6. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIOT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* **2023**, *23*, 5941. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Ullah, I.; Khalil, I.; Bai, X.; Garg, S.; Kaddoum, G.; Hossain, M.S. An Ensemble-Based Hybrid Model for the Detection of Attacks in the Internet of Vehicular Things. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 17914–17927. [\[CrossRef\]](#)
8. Aljabri, W.; Hamid, M.A.; Mosli, R. Lightweight and Adaptive Data-Driven Intrusion Detection System for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 2282–2292. [\[CrossRef\]](#)
9. Alharthi, M.; Medjek, F.; Djenouri, D. Ensemble Learning Approaches for Multi-Class Intrusion Detection Systems for the Internet of Vehicles (IoV): A Comprehensive Survey. *Future Internet* **2025**, *17*, 317. [\[CrossRef\]](#)

10. Yakan, H.; Fajjari, I.; Aitsaadi, N.; Adjih, C. Federated Learning for V2X Misbehavior Detection System in 5G Edge Networks. In Proceedings of the 26th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2023), Montreal, QC, Canada, 30 October–3 November 2023; pp. 155–163. [\[CrossRef\]](#)
11. Hamhoum, W.; Cherkaoui, S. MistralBSM: Leveraging Mistral-7B for Vehicular Networks Misbehavior Detection. *arXiv* **2024**, arXiv:2407.18462.
12. Sakr, H.A.; Fouda, M.M.; Ashour, A.F.; Abdelhafeez, A.; El-Afifi, M.I.; Abdellah, M.R. Machine Learning-Based Detection of DDoS Attacks on IoT Devices in Multi-Energy Systems. *Egypt. Inform. J.* **2024**, *28*, 100540. [\[CrossRef\]](#)
13. Sakr, H.A.; El-Afifi, M.I.; El-Mowafy, M.A.; Ibrahim, H.M. Detecting DDoS Threats in IoT-Driven 6G-Energy Hubs Networks Using Machine Learning Algorithms. *Discov. Appl. Sci.* **2025**, *7*, 1002. [\[CrossRef\]](#)
14. Sakr, H.A.; Eladl, A.A.; Ibrahim, H.M.; El-Afifi, M.I. Application of Machine Learning Techniques for Predicting False Data Injection Attacks in Smart Grid. In Proceedings of the 25th International Middle East Power System Conference (MEPCON), Cairo, Egypt, 17–19 December 2024; pp. 1–6. [\[CrossRef\]](#)
15. Garcia, M.H.C.; Molina-Galan, A.; Boban, M.; Gozalez, J.; Coll-Perales, B.; Şahin, T.; Kousaridas, A. A Tutorial on 5G NR V2X Communications. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1972–2026. [\[CrossRef\]](#)
16. He, Y.; Wang, D.; Huang, F.; Zhang, R.; Gu, X.; Pan, J. A V2I and V2V Collaboration Framework to Support Emergency Communications in ABS-Aided Internet of Vehicles. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 2038–2051. [\[CrossRef\]](#)
17. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
18. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63*, 3–42. [\[CrossRef\]](#)
19. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [\[CrossRef\]](#)
20. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [\[CrossRef\]](#)
21. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 3146–3154.
22. Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed.; Independently Published: Munich, Germany, 2022; ISBN 979-8-4114-6333-0.
23. Senevirathna, T.; La, V.H.; Marchal, S.; Siniarski, B.; Liyanage, M.; Wang, S. A Survey on XAI for 5G and Beyond Security: Technical Aspects, Challenges and Research Directions. *arXiv* **2022**, arXiv:2204.12822.
24. Hacking and Countermeasure Research Lab (HCRL). Car-Hacking Dataset. Korea University, 2018. Available online: <https://ocslab.hksecurity.net/Datasets/car-hacking-dataset> (accessed on 22 August 2026).
25. Matthews, B.W. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochim. Biophys. Acta* **1975**, *405*, 442–451. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [\[CrossRef\]](#)
27. Saito, T.; Rehmsmeier, M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE* **2015**, *10*, e0118432. [\[CrossRef\]](#) [\[PubMed\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.