SIFT-IDS: Sparse Iterative Feature Tuning for In-Vehicle Intrusion Detection

Congpu Wang, Jing Li* Weipeng Zeng, Mohammed N.M Ali, Burra Venkata Durga Kumar, Ahmad Affandi Supli*
Computing and Data Science
Xiamen University Malaysia
Sepang, Malaysia
{hugh.wangcongpu,clauvio.weipengzeng }@outlook.com
{jing.li,ali.mohammed,venkata.burra,ahmadaffandi}@xmu.edu.my

***Abstract*—Bit-level Controller Area Network (CAN) representations for in-vehicle intrusion detection are often high dimensional and redundant, which makes compact feature discovery essential for deployment under the resource constraints of the Internet of Vehicles (IoV). Existing feature-selection pipelines, however, often rank features using criteria or evaluators that differ from the classifier ultimately used for prediction, a discrepancy that may lead to a mismatch between subset search and downstream classification. To address this issue, we propose SIFT-IDS, a classifier-consistent recursive feature selection method that conducts feature ranking and final prediction within the same backbone family. Instantiated with FastKAN, SIFT-IDS derives lightweight architecture-derived feature scores from feature-indexed first-layer parameter groups, where scoring is performed by parameter extraction and norm evaluation without additional forward passes or auxiliary evaluators. Recursive re-ranking is further stabilized through first-layer regularization. On CICIoV2024, SIFT-IDS compresses a 75-dimensional CAN representation to 8–10 features while still maintaining six-class detection performance. Under the reported protocol, the 8-feature subset achieves 100% accuracy and 100% Macro-F1, reduces MACs by 82.7% (from 15,552 to 2,688), shrinks model size from 66.04 KB to 15.29 KB, and improves CPU throughput from 0.52M to 1.35M samples/s on our experimental CPU setup. Taken together, these results suggest that SIFT-IDS is an effective offline feature-selection method for learning compact CAN representations and stabilizes recursive re-ranking through first-layer L_1-induced sparsity, which promotes consistent feature discrimination under redundancy.**

***Keywords*-Cybersecurity, Internet of Vehicles, Intrusion Detection System, Machine Learning, Recursive Feature Elimination**

I. Introduction

The intrusion detection of Controller Area Network (CAN) becomes an indispensable tool for ensuring the safety of the Internet of Vehicles (IoV), since cyber attacks on IoVs can lead to dangerous consequences in terms of manipulating or injecting malicious frames [1], [2], [3], [4]. However, detection accuracy alone is not the only issue to consider in practical CAN monitoring applications. Specifically, the bit-level representation of CAN data is high dimensional, whereas the computing platform for detecting intrusions, such as in-vehicle processors or edge devices, suffers from constrained computing resources [5], [6], [7]. Thus, efficient feature learning becomes an important problem in deployment-friendly CAN intrusion detection systems.

A direct approach to alleviate the deployment overhead is the dimensionality reduction of CAN data before classification. In comparison with model-level lightweighting, the dimensionality reduction via feature selection eliminates the redundancy from the beginning of the processing pipeline and prevents unnecessary calculations on later nonlinear transformation [8], [9], [10]. Given the high dimensionality nature of bit-level CAN representations, where many features can be either irrelevant or highly correlated, this preprocessing technique can be beneficial for achieving both computational cost savings and increased detection performance [11], [12], [13].

Nonetheless, current feature-selection procedures are not always consistent with the final classifier adopted in deployments. Specifically, it is common practice to evaluate the feature importance by statistical tests, filter scores, decision trees, or other auxiliary methods while the final classifier remains a different nonlinear model [8], [9], [14], [15], [16]. As a result, there can be some inconsistency between the subset searching phase and the following classification stage.

To address this issue, we propose SIFT-IDS, a classifier-consistent recursive feature selection method in which feature ranking and final prediction are performed within the same backbone family. Rather than relying on an external evaluator, SIFT-IDS repeatedly retrains the classifier on the current subset and updates feature relevance after each elimination round, which keeps subset discovery aligned with the reduced classifier used at that stage. In this work, SIFT-IDS is instantiated with FastKAN, whose grouped first-layer structure enables lightweight architecture-derived feature scoring. We further introduce first-layer L_1 regularization to stabilize recursive re-ranking under correlated and redundant inputs. In this context, "classifier-consistent" denotes continuity within the same backbone family across subset search and final prediction, rather than a formal statistical consistency guarantee.

We apply SIFT-IDS to feature selection and detection of CAN intrusion in CICIoV2024. Our results demonstrate that, in the proposed framework, the 75-dimensional input CAN space can be effectively compressed to 8-10 dimensional yet highly discriminative subspaces without any decrease in six-class classification accuracy and significant improvement in deployment efficiency. The main contributions of this paper

include the following:

- **Classifier-consistent recursive feature selection.** We introduce a recursive subset-discovery scheme in which feature ranking and final prediction are performed within the same backbone family, reducing the mismatch between feature selection and downstream classification. In this work, classifier consistency refers to procedural alignment during subset discovery and deployment, rather than a formal statistical consistency guarantee.
- **Architecture-derived lightweight feature scoring.** We propose a feature-indexed first-layer path-energy score that can be computed directly from trained parameters without additional validation passes, feature masking, or auxiliary feature evaluators.
- **Compact feature discovery for CAN intrusion detection.** On CICIoV2024, our method selects compact subsets of 8–10 dimensions from a 75-dimensional input while maintaining saturated six-class detection performance and substantially reducing deployment-stage computational cost.

II. Methodology

A. Overview

We propose SIFT-IDS, a classifier-consistent recursive feature selection method for identifying compact and discriminative subsets from high-dimensional CAN representations. The key idea is to keep feature ranking within the same backbone family used for final prediction, so that subset discovery remains aligned with downstream classification. Here, classifier-consistent refers to procedural continuity within the same backbone family.

Starting from the initial feature set F_0, SIFT-IDS iteratively trains the backbone on the current subset F_t, computes feature scores directly from the trained model, removes the lowest-ranked features to obtain F_{t+1}, and then retrains the backbone on the reduced subset. This process continues until the validation trajectory indicates that further elimination is no longer beneficial. The best subset is selected according to validation Macro-F1, after which the classifier is retrained once on the selected features for deployment. Figure 1 summarizes the overall SIFT-IDS pipeline for CAN intrusion detection.

This design targets four practical requirements:

- Ranking should be generated by the same model family used at deployment.
- Scoring should incur minimal overhead.
- Recursive re-ranking should remain stable across elimination rounds.
- Input reduction should translate into lower deployment cost.

B. FastKAN-Based Architecture-Derived Feature Scoring

To realize lightweight classifier-consistent ranking, the backbone should expose a feature-indexed structure that supports efficient per-feature scoring. We instantiate SIFT-IDS with FastKAN, which preserves grouped first-layer pathways while replacing spline evaluation with a more efficient radial basis function (RBF) approximation[17], [18].

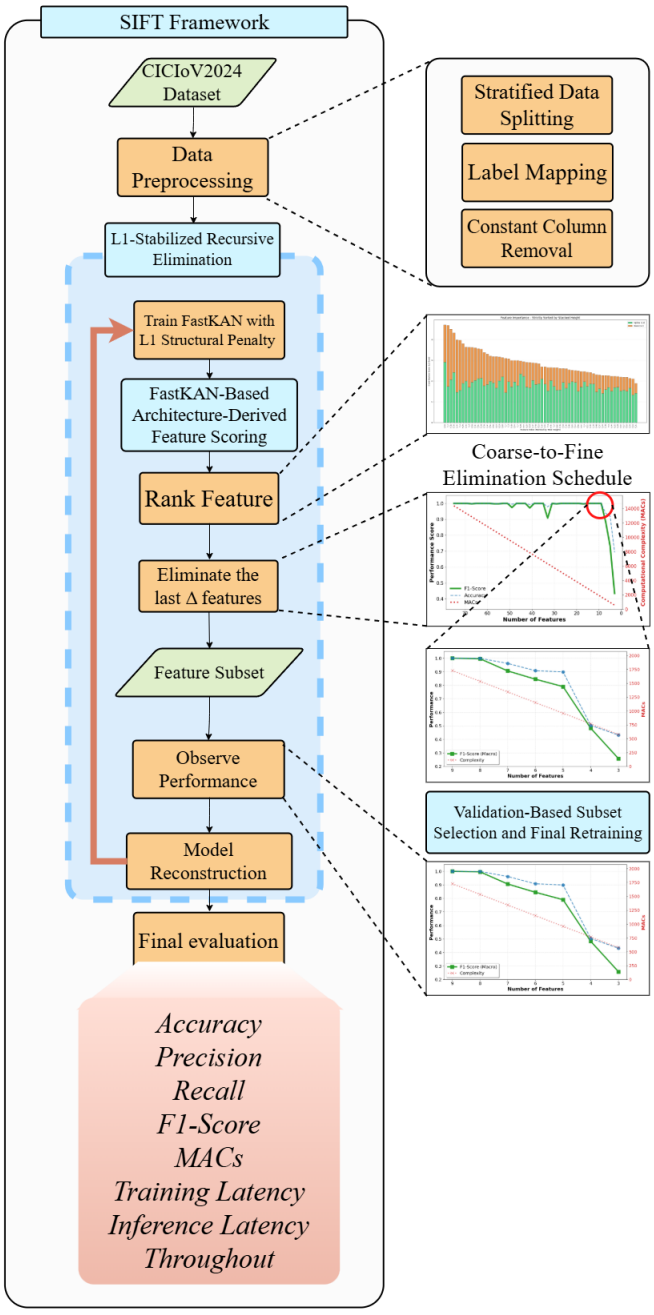


Fig. 1. SIFT-IDS Framework for IoV Intrusion Detection.

For a retained feature i, let m denote the hidden width and g the RBF grid size. In the first layer, feature i is associated with a linear branch and an RBF-based nonlinear branch. We group its first-layer parameters as

$$\Theta_i^{(1)} = \left\{ w_{j,i}^{(b)} \right\}_{j=1}^{m} \cup \left\{ w_{j,i,k}^{(\mathrm{rbf})} \right\}_{j=1,k=1}^{m,g}, \tag{1}$$

where $w_{j,i}^{(b)}$ is the linear-branch weight from feature i to the j-th hidden unit, and $w_{j,i,k}^{(\mathrm{rbf})}$ is the coefficient of the k-th RBF

basis associated with feature i and hidden unit j.

Based on this grouped representation, the relevance score of feature i is defined as the first-layer path energy

$$S_i = \sqrt{\sum_{j=1}^{m} \left(w_{j,i}^{(b)}\right)^2 + \sum_{j=1}^{m}\sum_{k=1}^{g} \left(w_{j,i,k}^{(\mathrm{rbf})}\right)^2}. \tag{2}$$

A larger S_i indicates stronger first-layer participation of feature i in the current classifier. Since S_i is computed directly from trained parameters, scoring requires only parameter extraction and norm evaluation, without additional forward passes or auxiliary evaluators. Specifically, after the FastKAN model is trained at a given elimination round, SIFT-IDS extracts the linear-branch and RBF-branch parameters associated with each retained input feature and computes their grouped Euclidean norm. Unlike wrapper-based evaluators that repeatedly mask features or measure validation performance after removing individual dimensions, the proposed scoring step does not require additional inference over the training or validation samples. Therefore, the ranking cost depends mainly on the number of retained features, hidden units, and RBF bases, rather than on the number of data samples. In this work, S_i is used as a lightweight architecture-derived relevance proxy rather than a formal feature-importance measure[18]. This sparsity-aware scoring implicitly suppresses weak feature pathways and improves the robustness of ranking under correlated inputs.

C. L_1-Stabilized Recursive Elimination

Because recursive elimination repeatedly retrains the classifier on changing subsets, feature scores may fluctuate across rounds, especially under correlated or redundant inputs. To improve ranking stability and encourage sparse feature utilization, SIFT-IDS applies a first-layer L_1 penalty during elimination:

$$\mathcal{L}_{\mathrm{elim}} = \mathcal{L}_{\mathrm{WCE}} + \lambda\,\Omega(W), \tag{3}$$

where $\mathcal{L}_{\mathrm{WCE}}$ is class-weighted cross-entropy, λ controls sparsity strength, and $\Omega(W)$ denotes the L_1 penalty on all first-layer parameters involved in scoring. Further details are set out in Equation (7) in the Appendix.

At round t, SIFT-IDS trains FastKAN on F_t, computes S_i for all $i \in F_t$, removes the Δ_t lowest-ranked features, and rebuilds the input layer for the next round. To reduce offline search cost, we use a coarse-to-fine schedule: larger pruning steps are adopted when the subset is still large, and $\Delta_t = 1$ is used in the final stage. In our implementation, $\Delta_t = 2$ in the coarse stage and $\Delta_t = 1$ in the fine stage.

Because both retraining and re-scoring are performed after each pruning step, the elimination trajectory remains aligned with the current reduced classifier rather than with the initial full-dimensional model.

D. Validation-Based Subset Selection and Final Retraining

SIFT-IDS does not assume a predefined target dimensionality. Instead, subset selection is based on the validation trajectory over the development pool. We use Macro-F1 as the primary criterion due to class imbalance. Let M_t denote the validation Macro-F1 at round t. The search stops after the best observed value is followed by two consecutive declines, and the final subset is chosen as the one achieving the highest validation Macro-F1, with ties broken in favor of the smaller subset.

After the subset is fixed, the deployment-stage classifier is retrained on the selected features without the sparsity term:

$$\mathcal{L}_{\mathrm{final}} = \mathcal{L}_{\mathrm{WCE}}. \tag{4}$$

Thus, $\mathcal{L}_1$ regularization is used only to stabilize recursive ranking during subset search, not to constrain the final deployed detector. Overall, SIFT-IDS acts as an offline subset-discovery procedure that couples recursive feature selection with the deployed classifier family and produces compact discriminative feature subsets for CAN intrusion detection under deployment constraints.

E. Rationale and Scope of the Proposed Score

The proposed scoring mechanism is designed as a classifier-aligned relevance proxy rather than a formal causal feature-importance measure. Its rationale is based on the feature-indexed first-layer structure of FastKAN. For each retained input dimension, FastKAN maintains a dedicated group of first-layer linear and RBF parameters. When this parameter group has a larger norm after training, the corresponding feature has stronger direct participation in the learned representation before subsequent nonlinear transformations. The classifier-consistent design further reduces the mismatch between feature selection and final prediction. In many feature-selection pipelines, features are ranked by external statistical criteria, tree-based estimators, or auxiliary models, while the final classifier belongs to a different model family. In contrast, SIFT-IDS ranks features using the same backbone family that is later used for deployment. As a result, the subset search is aligned with the inductive bias of the deployed classifier. First-layer L1 regularization is used to stabilize this recursive process. Under correlated and redundant inputs, multiple features may provide similar discriminative information, causing their ranking positions to fluctuate across elimination rounds. The L1 penalty encourages weakly used feature pathways to shrink toward zero, making competition among redundant dimensions more explicit. This helps recursive re-ranking progressively remove weak or replaceable features while preserving compact discriminative subsets. Nevertheless, the proposed score should not be interpreted as a theoretical guarantee of global optimality. Because SIFT-IDS performs greedy recursive elimination, the selected subset may still be path-dependent. The goal of the proposed score is to provide a lightweight, classifier-aligned, and deployment-oriented criterion for compact feature discovery.

III. Experimental Setup

A. Dataset, Preprocessing, and Evaluation Protocol

SIFT-IDS is evaluated on CICIoV2024, a CAN intrusion detection dataset collected from direct attacks on messages

injected into the in-vehicle CAN bus. Each raw CAN message contains an Arbitration ID and up to eight payload bytes. In this work, these fields are converted into a fixed-length bit-level representation, where each bit position is treated as one candidate feature. Concretely, each bit derived from the Arbitration ID and payload bytes is treated as an input dimension, and each resulting bit-level vector is assigned one traffic label, such as BENIGN or one of the predefined attack categories. Redundant feature dimensions are removed in two stages. First, zero-variance columns are discarded before model training because their values remain constant across all samples and therefore cannot contribute to class discrimination. This preprocessing step removes 78 zero-variance columns from the provided 153-dimensional representation, resulting in a 75-dimensional candidate feature space. Second, SIFT-IDS further performs task-driven recursive elimination on these 75 retained dimensions to remove weakly participating or redundant features according to the architecture-derived path-energy score.

Following the class assignment, the original *specific_class* is mapped to a six-class problem: BENIGN, DoS, GAS, RPM, SPEED, and STEERING_WHEEL. In order to ensure a leak-proof testing scheme, we initially set aside a stratified 15% test set and exclude it from any selection or training process; all selections and evaluations for recursive elimination are solely based on the remaining 85% development set.

Since the class distribution is imbalanced, Macro-F1 is used as the primary evaluation metric because it gives equal importance to each class and is less dominated by majority categories. Additional evaluation metrics include accuracy, precision, and recall. For efficiency analysis, we report MACs, model size, latency per sample on the CPU, and throughput. Figure 2 shows the class distribution of CICIoV2024 after mapping the original labels into six categories. The distribution is imbalanced across benign and attack classes, which motivates the use of class-weighted cross-entropy during training and Macro-F1 as the primary evaluation metric.

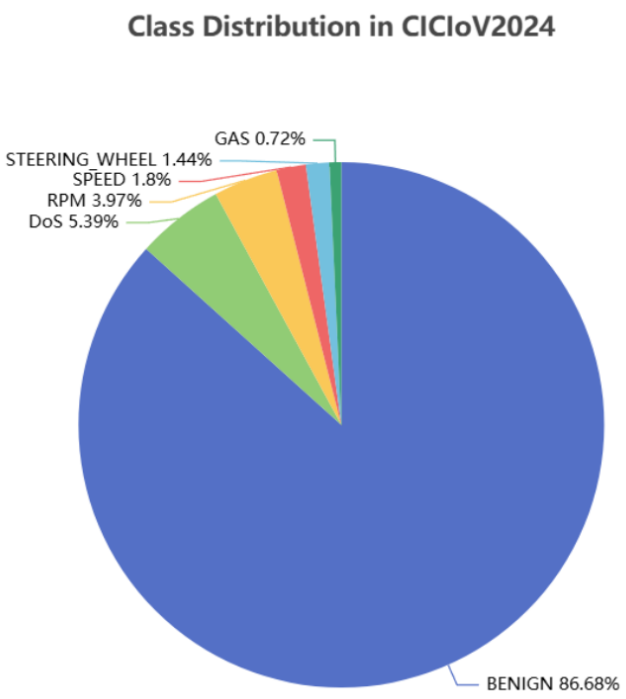


Fig. 2. Class Distribution in CICIoV2024.

B. Backbone and Baselines

FastKAN is adopted as the unified backbone architecture in our recursive feature selection framework and final classifier. Based on preliminary experimentation on the development dataset, we use a compact configuration with hidden width 32 and grid size 5; the input size depends on the subset selected during elimination. During the whole training procedure, we adopt class-balanced cross-entropy and apply L_1 regularization to the first-layer weights during recursive elimination only.

Specifically, we test five configurations:

- **A1:** FastKAN without feature selection using all features;
- **A2:** FastKAN with one-shot feature selection and no L_1;
- **A3:** FastKAN with one-shot selection and L_1;
- **A4:** FastKAN with recursive elimination and no L_1; and
- **A5:** SIFT-IDS with recursive elimination and L_1.

All detection results presented below are averaged over five seeds, unless specified otherwise. All experiments are performed in a CPU-only setting using AMD Ryzen 7 5800H, 16 GB RAM, and PyTorch 2.10.0.

IV. Results

A. Effectiveness and Ablation of the Proposed Feature Selection Method

Table I reports the ablation results. Overall, the most compact subset is obtained only when recursive re-ranking is combined with first-layer stabilization. The comparison among A2, A3, A4, and A5 also shows that feature-selection performance depends not only on whether ranking is performed once or recursively, but also on whether the ranking process is regularized to suppress unstable or weak feature pathways. The full-feature baseline A1 uses all 75 inputs and achieves 100% on all reported metrics. Static one-shot elimination already removes substantial redundancy: A2 reduces the input to 11 features with a slight drop in accuracy and Macro-F1, while adding first-layer L_1 in A3 further reduces the subset to 9 features and nearly restores full performance.

TABLE I
ABLATION OF SIFT-IDS COMPONENTS

Exp ID	Selection Strategy	L_1	Feature Dim.	Accuracy	Precision	Recall	Macro-F1
A1	None	×	75	100%	100%	100%	100%
A2	Fixed Ranking Elimination	×	11	99.9441%	99.9463%	99.9441%	99.9447%
A3	Fixed Ranking Elimination	✓	9	99.9991%	99.9991%	99.9991%	99.9991%
A4	RFE	×	13	99.9441%	99.9463%	99.9441%	99.9447%
A5	RFE	✓	8	100.0%	100.0%	100.0%	100.0%

TABLE II
EFFICIENCY COMPARISON UNDER FULL-FEATURE, SIFT-IDS-SELECTED, AND AUXILIARY DEVELOPMENT-SET SETTINGS

Model structure	Dataset	Model size	MACs	Training delay (s)	Inference delay (μs)	Throughput (samples/s)
75-32-6	Original	66.04 KB	15,552	121.19 ± 1.25	1.95 ± 0.33	$523{,}162.02 \pm 63{,}984.19$
75-32-6	D5	66.04 KB	15,552	12.28 ± 0.09	1.91 ± 0.25	$529{,}333.27 \pm 52{,}617.82$
8-32-6	Original	15.29 KB	2,688	101.76 ± 0.49	0.76 ± 0.12	$1{,}345{,}378.02 \pm 162{,}919.27$
10-32-6	D5	16.79 KB	3,072	10.06 ± 0.11	0.78 ± 0.11	$1{,}296{,}911.98 \pm 44{,}997.23$

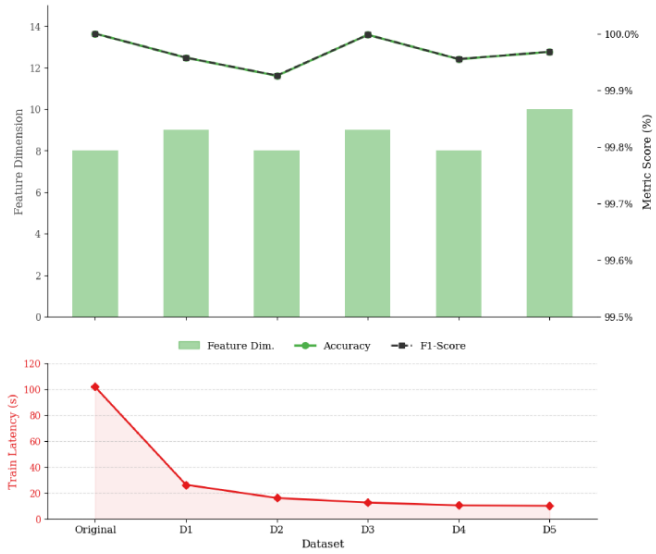


Fig. 3. SIFT-IDS performance across controlled development-set variants.

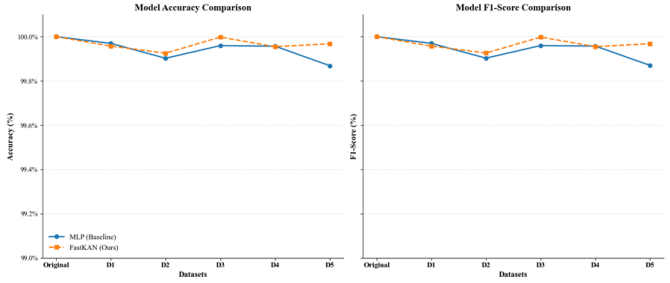


Fig. 4. Cross-model transferability of SIFT-IDS-selected feature subsets.

Recursive elimination alone does not improve the compactness–performance trade-off: A4 retains 13 features and remains slightly below the full model. In contrast, A5 combines recursive re-ranking with first-layer stabilization and reduces the input to 8 features while preserving 100% accuracy and 100% Macro-F1 under the reported protocol. This indicates that the gain critically depends on sparsity-induced stabilization provided by the L_1 term, which enables recursive re-ranking to progressively suppress redundant or weakly participating features while preserving discriminative ones. In other words, the benefit of SIFT-IDS is not merely caused by reducing the number of input dimensions, but by aligning recursive subset discovery with the classifier that is retrained at each elimination round.

Note: The reported feature dimensionality for each variant corresponds to the subset selected by the validation-based elimination trajectory on the development pool. Test-set metrics are reported only for final post hoc evaluation. The "$\geq$99.9%" statement is descriptive of the observed outcomes rather than a test-time selection criterion.

B. Robustness of Subset Discovery and Cross-Model Transferability

To examine whether the selected subsets are overly tied to a particular development-set composition, we rerun SIFT-IDS on five controlled development-set variants (D1–D5) with imbalance ratios ranging from 20 to 1, while keeping the same held-out test set unchanged. These controlled variants are used as a lightweight robustness assessment under different imbalance conditions and development-set compositions. Across these variants, SIFT-IDS consistently selects compact subsets of 8–10 features and maintains test accuracy and Macro-F1 above 99.92%. At the same time, the total offline search time decreases substantially, from 101.76 ± 0.49 s on the original development pool to 10.06–26.30 s on the adjusted variants.

We further transfer the selected subsets to a standard MLP by changing only the input dimensionality. The original 8-feature subset still achieves 100% test performance with the MLP, and Macro-F1 remains above 99.87% across D1–D5. Although this does not imply that all selected subsets are backbone-invariant, it suggests that SIFT-IDS captures generally discriminative information rather than only FastKAN-specific feature preferences. These results do not replace evaluation on fully independent CAN intrusion datasets, but

they provide initial evidence that the selected compact subsets remain effective under controlled distributional changes and classifier changes. Figure 3 reports the performance of SIFT-IDS across controlled development-set variants. Although the imbalance ratios and development-set compositions are changed, SIFT-IDS consistently selects compact subsets of 8–10 features and maintains high test performance. This indicates that the selected subsets are not solely tied to one specific development-set composition. Figure 4 evaluates the cross-model transferability of the SIFT-IDS-selected feature subsets. The selected subsets are transferred to a standard MLP by changing only the input dimensionality. The resulting performance remains high, suggesting that the discovered features contain generally discriminative information rather than only FastKAN-specific preferences.

C. Deployment Benefits of the Selected Compact Feature Subsets

Table II summarizes the deployment benefit of feature compression. Relative to the full-feature baseline [75-32-6], the compact detector [8-32-6] reduces MACs from 15,552 to 2,688 (-82.7%) and model size from 66.04 KB to 15.29 KB (-76.8%). CPU single-sample inference latency decreases from 1.95 ± 0.33 μs to 0.76 ± 0.12 μs, while throughput increases from 0.52M to 1.35M samples/s.

These results show that the main deployment gain comes from input-space compression itself. Since fewer input dimensions are processed by the first layer, the compact model requires fewer multiply–accumulate operations and stores fewer first-layer parameters. By contrast, the reported training time mainly reflects the offline cost of subset discovery and final retraining rather than deployment-stage overhead. SIFT-IDS should therefore be understood as an offline feature-subset discovery framework that produces a smaller and faster detector for resource-constrained in-vehicle deployment.

V. Discussion

These results indicate that SIFT-IDS functions not merely as a compression mechanism, but as a feature-selection framework that identifies compact and discriminative CAN representations. Static ranking can already remove substantial redundancy from bit-level CAN representations. However, recursive elimination alone does not necessarily guarantee the best compactness–performance trade-off, as shown by the comparison between A4 and A5. In contrast, recursive re-ranking combined with first-layer stabilization reduces the feature subset to the most compact configuration observed in our experiments while maintaining saturated detection performance under the reported protocol. In doing so, we may see that sparsity itself is a structural constraint that enables features to compete with each other recursively.

From an applied point of view, the primary value of SIFT-IDS lies in the offline identification of compact feature subsets for deployment-restricted CAN detection. Although the recursive approach introduces additional training-stage complexity, the reduction in the number of features continuously benefits online processing efficiency in terms of computation, memory footprint, and CPU performance. Moreover, the results on stability and transferability suggest that the algorithm identifies generic features that distinguish among classes rather than merely learning backbone-specific preferences.

The controlled development-set variants and cross-model transfer experiment provide initial robustness evidence, but they should not be interpreted as a complete substitute for cross-dataset validation. Since different CAN intrusion datasets may use different vehicles, message distributions, attack implementations, and feature representations, additional independent datasets are still necessary to fully assess dataset-level generalization. In this work, we focus on CICIoV2024 and evaluate robustness through controlled imbalance variants and classifier transfer, while leaving broader cross-dataset validation for future work.

VI. Threats to Validity and Limitations

Several limitations should be noted. First, SIFT-IDS employs greedy recursive elimination, and therefore subset discovery is path-dependent. Minor differences in early pruning decisions, caused for example by random initialization or sampling variation, may lead to different yet similarly effective subsets. Second, the proposed path-energy score is an architecture-derived relevance proxy rather than a formal causal or globally optimal feature-importance measure. Although the score reflects the first-layer participation of each retained feature in the trained FastKAN classifier, it may not fully capture high-order feature interactions formed in deeper nonlinear transformations. Third, the experiments are conducted on CICIoV2024 under the described split protocol and CPU-only environment. Although controlled development-set variants and cross-model transfer experiments are included to assess robustness, evaluation on additional independent CAN intrusion datasets is still required to further verify dataset-level generalization. Fourth, although the held-out test set is isolated before feature selection and model training, a negligible amount of overlap between different partitions may still exist in CAN traffic due to the characteristics of message-level data. Finally, SIFT-IDS improves deployment-stage inference efficiency, but its offline subset-discovery process is more computationally intensive than one-shot feature selection because it relies on iterative retraining.

VII. Conclusion and Future Work

This paper described SIFT-IDS, a classifier-consistent recursive feature selection method for compact CAN intrusion detection under IoV resource limitations. Through recursive subset discovery and integration with the deployed classifier family via architecture-inspired path-energy scoring, combined with stabilization during elimination using first-layer L_1 regularization, the method compressed a 75-dimensional bit representation of CAN messages into 8–10 features without sacrificing near-maximal six-class detection performance on CICIoV2024. Beyond the specific IDS application, the

findings demonstrate that classifier-consistent recursive feature ranking provides a promising approach for identifying compact discriminative subsets of structured data. Future research will involve validation on additional independent CAN intrusion datasets, on-device implementation, improvement of recursive search efficiency, and adaptation of the technique to additional tabular security tasks. An anonymized repository that includes part of the project's code will be released upon publication https://github.com/N1ne-W/SIFT-IDS.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of the Xiamen University Malaysia Research Fund (XMUMRF), Cycle 17/2026, under Grant XMUMRF/2026-C17/IECE/0071 and XMUMRF/2022-C9/IECE/0031.

REFERENCES

[1] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cyberattacks and countermeasures for in-vehicle networks," vol. 54, no. 1, pp. 1–37. [Online]. Available: https://dl.acm.org/doi/10.1145/3431233

[2] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, and S. Yu, "Attacks and defences on intelligent connected vehicles: a survey," vol. 6, no. 4, pp. 399–421. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235286481930197X

[3] D. C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," pp. 1–91.

[4] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, G. Madzudzo, and M. Cheah, "AI-based intrusion detection systems for in-vehicle networks: a survey," vol. 55, no. 11, pp. 237:1–237:40. [Online]. Available: https://dl.acm.org/doi/10.1145/3570954

[5] T. Gong, L. Zhu, F. R. Yu, and T. Tang, "Edge intelligence in intelligent transportation systems: a survey," vol. 24, no. 9, pp. 8919–8944. [Online]. Available: https://ieeexplore.ieee.org/document/10133894/

[6] A. Haddaji, S. Ayed, and L. Chaari Fourati, "IoV security and privacy survey: issues, countermeasures, and challenges," vol. 80, no. 15, pp. 23 018–23 082. [Online]. Available: https://doi.org/10.1007/s11227-024-06269-5

[7] C. Li, M. Dong, Y. Fu, F. Richard Yu, and N. Cheng, "Integrated sensing, communication, and computation for IoV: challenges and opportunities," vol. 28, pp. 1136–1168. [Online]. Available: https://ieeexplore.ieee.org/document/11175176/

[8] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A review of feature selection methods for machine learning-based disease risk prediction," vol. 2, pp. 1–17. [Online]. Available: https://www.frontiersin.org/journals/bioinformatics/articles/10.3389/fbinf.2022.927312/full

[9] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," vol. 19, no. 1, pp. 3–26. [Online]. Available: https://www.sciendo.com/article/10.2478/cait-2019-0001

[10] Z. S. Mahdi, R. M. Zaki, and L. Alzubaidi, "Advanced hybrid techniques for cyberattack detection and defense in IoT networks," vol. 8, no. 2, p. e471. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/spy2.471

[11] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," vol. 12, no. 10, p. 5015. [Online]. Available: https://www.mdpi.com/2076-3417/12/10/5015

[12] M. B. Musthafa, S. Huda, Y. Kodera, M. A. Ali, S. Araki, J. Mwaura, and Y. Nogami, "Optimizing IoT intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques," vol. 24, no. 13, p. 4293. [Online]. Available: https://www.mdpi.com/1424-8220/24/13/4293

[13] S. Walling and S. Lodh, "Enhancing IoT intrusion detection through machine learning with AN-SFS: a novel approach to high performing adaptive feature selection," vol. 4, no. 1, pp. 16–41. [Online]. Available: https://link.springer.com/10.1007/s43926-024-00074-5

[14] A. Sharma and M. Singh, "Batch reinforcement learning approach using recursive feature elimination for network intrusion detection," vol. 136, p. 109013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0952197624011710

[15] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning," vol. 11, no. 1, pp. 133–164. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00994-7

[16] R. Saia, S. Carta, G. Fenu, and L. Pompianu, "Leveraging the training data partitioning to improve events characterization in intrusion detection systems," vol. 14, no. 6, pp. 1345–1353. [Online]. Available: https://www.jait.us/show-233-1448-1.html

[17] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: kolmogorov-arnold networks." [Online]. Available: http://arxiv.org/abs/2404.19756

[18] A.-C. Akazan and V. R. Mbingui, "Splines-based feature importance in kolmogorov-arnold networks: a framework for supervised tabular data dimensionality reduction." [Online]. Available: http://arxiv.org/abs/2509.23366

APPENDIX

For completeness, this appendix summarizes the core formulation of SIFT-IDS used in the FastKAN instantiation. The held-out test set is never used during subset search. Recursive elimination is performed only on the development set.

Let F_t denote the retained feature subset at elimination round t, with $d_t = |F_t|$, and let Δ_t denote the number of features removed at round t. In our coarse-to-fine search, $\Delta_t = 2$ in the coarse stage and $\Delta_t = 1$ in the final fine-grained stage.

For each retained feature $i \in F_t$, SIFT-IDS obtains the relevance score S_i from the feature-indexed first-layer parameter group of FastKAN, following the path-energy definition in Section II-B. This score is used only for ranking retained features during recursive elimination. The remaining appendix summarizes how the elimination loss, subset update, and final retraining objective are applied after the scores are obtained.

During recursive elimination, the model is trained with class-weighted cross-entropy and first-layer sparsity regularization:

$$\mathcal{L}_{\text{elim}} = \mathcal{L}_{\text{WCE}} + \lambda \sum_{i \in F_t} \left(\sum_{j=1}^{m} \left| w_{j,i}^{(b)} \right| + \sum_{j=1}^{m} \sum_{k=1}^{g} \left| w_{j,i,k}^{(\text{rbf})} \right| \right). \tag{5}$$

After training on F_t, features are ranked by S_i, and the next subset is updated by removing the Δ_t lowest-ranked features:

$$F_{t+1} = F_t \setminus \mathcal{R}_t, \qquad |\mathcal{R}_t| = \Delta_t. \tag{6}$$

The final subset $F^\star$ is selected from the validation trajectory using Macro-F1, with ties resolved in favor of the smaller subset. After subset selection, the deployment-stage classifier is retrained on $F^\star$ without the sparsity term:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{WCE}}. \tag{7}$$

Thus, SIFT-IDS remains classifier-consistent throughout the pipeline in the procedural sense: feature ranking, recursive re-ranking, and final classification are all carried out within the same FastKAN backbone family. This statement does not

imply a formal statistical consistency guarantee, but emphasizes that subset discovery and deployment-stage prediction are aligned through the same classifier family.