STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, May 2026, pp 4437–4463.
Published online in International Academic Press (www.IAPress.org)

MSDA-GDS: A Dual-Branch Hybrid Federated Explainable Deep Learning Framework for CAN Bus Intrusion Detection in Internet of Vehicles

Moh'D Suliman Shakkah[1,*], Belal Al-sellami[2], Abdulnaser A. Hagar[3], Mohammed Tawfik[4]

[1]*Department of Computer Science, Faculty of Science and Information Technology, Irbid National University, Irbid, JORDAN*
[2]*Department of Computer Science, Dr. Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar, Maharashtra, INDIA*
[3]*Faculty of Data Science and Information Technology, INTI International University, Nilai 71800, Negeri Sembilan, MALAYSIA*
[4]*Department of Cyber Security, Faculty of Information Technology, Ajloun National University, P.O.43, Ajloun-26810, JORDAN*

Abstract The Controller Area Network (CAN) bus remains critically vulnerable to cyberattacks due to its lack of authentication and encryption. Existing intrusion detection systems (IDS) for Internet of Vehicles (IoV) suffer from single-branch architectures that fail to capture multi-scale CAN byte dependencies, centralized training paradigms that compromise vehicular data privacy, and insufficient model interpretability. This paper proposes MSDA-GDS, a dual-branch hybrid federated explainable framework comprising a Multi-Scale Dilated Attention (MSDA) branch with parallel dilated convolutions and channel-spatial attention, and a Gated Depthwise Separable (GDS) branch with learnable gating mechanisms and residual connections, fused via learned attention weighting. The framework integrates Apache Spark-accelerated preprocessing, FedProx federated learning with differential privacy, and multi-method explainability (SHAP, LIME, gradient saliency). Evaluation on CICIoV2024 (1,408,219 CAN frames) and CIC-IDS-2017 (2.83M flows) demonstrates 99.99% and 99.40% accuracy respectively, with the federated variant achieving 99.97% under full privacy protection. Ablation analysis confirms the gating mechanism (ΔF1 = -0.21) and engineered features (ΔF1 = -0.27) as the most impactful components, while XAI analysis identifies DATA_2, DATA_1, and DATA_3 as the most discriminative byte positions with high cross-method consistency ($\rho = 0.978$). These contributions collectively advance the development of secure and sustainable intelligent transportation systems.

Keywords Intrusion Detection System; Controller Area Network; Federated Learning; Differential Privacy; Multi-Scale Dilated , Gated Depthwise Convolutions; Explainable AI (SHAP).

DOI: 10.19139/soic-2310-5070-3599

1. Introduction

The rapid proliferation of connected and autonomous vehicles (CAVs) has fundamentally transformed the modern transportation landscape, enabling advanced functionalities such as cooperative driving, over-the-air diagnostics, and vehicle-to-everything (V2X) communication [1, 7]. At the core of intra-vehicular communication lies the Controller Area Network (CAN) bus protocol, originally designed in the 1980s for reliable real-time communication among Electronic Control Units (ECUs) without inherent security mechanisms [3, 4]. The absence of authentication, encryption, and access control in the CAN protocol renders modern vehicles critically vulnerable to a spectrum of cyberattacks, including Denial-of-Service (DoS) flooding, message spoofing (targeting gas, RPM, speed, and steering wheel sensors), replay attacks, and data falsification [2, 5]. With the global connected vehicle market projected to exceed 400 million units by 2030, the development of robust, privacy-preserving, and interpretable Intrusion Detection Systems (IDS) for vehicular networks has become an imperative research priority.

*Correspondence to: Moh'D Suliman Shakkah (Email: m.shakkah@inu.edu.jo). Department of Computer Science, Faculty of Science and Information Technology, Irbid National University, Irbid, JORDAN.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Traditional machine learning approaches, including Random Forest, XGBoost, and Support Vector Machines, have demonstrated considerable success in CAN bus intrusion detection, achieving near-perfect accuracy on standardized benchmarks [1, 2, 15, 31]. However, these approaches exhibit several fundamental limitations: (i) reliance on centralized data collection that violates vehicular data privacy regulations, (ii) inability to capture complex multi-scale spatial dependencies inherent in CAN bus byte patterns, (iii) limited feature extraction capability that fails to simultaneously model local byte-level anomalies and global cross-byte attack signatures, and (iv) lack of model interpretability essential for safety-critical automotive deployments [7, 8]. Deep learning architectures, including CNN-GRU hybrids [7], sequence autoencoders with multi-head attention [6], and transformer-based zero-shot frameworks [8], have advanced detection performance but typically operate as single-branch architectures that capture either spatial or temporal features independently, without jointly modeling multi-scale patterns and efficient gated representations.

Federated Learning (FL) has emerged as a promising paradigm for distributed vehicular security, enabling multiple vehicles or edge nodes to collaboratively train intrusion detection models without sharing raw CAN bus data [9, 10, 11, 35, 36]. Nevertheless, existing FL-based IDS predominantly employ conventional architectures (RNNs, XGBoost ensembles) within basic FedAvg aggregation, lacking robustness to non-IID data distributions and formal privacy guarantees. Furthermore, the integration of distributed computing frameworks such as Apache Spark for scalable preprocessing with federated training pipelines remains underexplored in the vehicular security domain [12, 13, 32].

Explainable Artificial Intelligence (XAI) has become increasingly critical for deploying IDS in safety-critical vehicular environments, where understanding *why* a CAN frame was classified as malicious is as important as the classification itself [7, 14]. While SHAP-based post-hoc explanations have been applied to CNN-GRU models [7], comprehensive multi-method XAI analysis combining global (SHAP), local (LIME), and gradient-based (saliency maps) interpretability within a federated vehicular IDS framework has not been investigated [33, 38].

To address these challenges, this paper proposes **MSDA-GDS**, a novel dual-branch hybrid federated explainable deep learning framework for Internet of Vehicles intrusion detection. The key contributions are as follows:

1. **Novel Dual-Branch Hybrid Architecture (MSDA-GDS):** We propose a unified model with two complementary branches: the Multi-Scale Dilated Attention (MSDA) branch employing parallel dilated convolutions ($d \in \{1, 2, 4\}$) with channel-spatial attention, and the Gated Depthwise Separable (GDS) branch utilizing learnable gating mechanisms ($\tanh \odot \sigma$) with residual connections, fused via learned attention-weighted integration for parameter-efficient CAN bus intrusion detection.

2. **Privacy-Preserving Federated Learning:** We integrate FedProx aggregation with proximal regularization and (ε, δ)-differential privacy via calibrated Gaussian noise injection, enabling collaborative model training across distributed vehicular clients without sharing raw CAN bus data, with Apache Spark-accelerated preprocessing for scalable feature engineering.

3. **Comprehensive Explainability and Dual-Dataset Evaluation:** We provide multi-method XAI analysis (SHAP, LIME, gradient saliency) for model transparency, validated through extensive experiments on CICIoV2024 (1,408,219 CAN frames) and CIC-IDS-2017 (2.83M network flows), achieving 99.99% and 99.40% accuracy respectively with systematic ablation analysis.

The remainder of this paper is organized as follows: Section 2 reviews the related literature across CAN bus IDS, federated learning, deep learning architectures, and XAI approaches. Section 3 details the proposed MSDA-GDS framework architecture, federated training protocol, and XAI methodology. Section **??** presents the experimental setup, datasets, and implementation details. Section 4 reports and analyzes the experimental results. Section 5 discusses the findings, limitations, and implications. Finally, Section 6 concludes the paper with future research directions.

2. Related Work

The proliferation of connected and autonomous vehicles has intensified research efforts in Controller Area Network (CAN) bus intrusion detection systems. This section comprehensively reviews the existing literature across five interconnected domains: traditional and ensemble machine learning-based IDS for vehicular networks, deep learning and attention-based architectures for CAN traffic analysis, federated learning approaches for privacy-preserving intrusion detection, explainable AI techniques for security applications, and emerging frameworks leveraging distributed computing. We critically analyze the strengths, limitations, datasets employed, and performance metrics of prior work to contextualize our contributions and establish benchmarks for comparative evaluation.

2.1. Machine Learning-Based CAN Bus Intrusion Detection

Traditional machine learning approaches have established strong baselines for CAN bus intrusion detection. Janbi [1] conducted a comprehensive evaluation of 25 machine learning models for AI-driven intrusion detection in Internet of Vehicles (IoV) communications. The study leveraged the CICIoV2024 dataset comprising 1,408,219 original CAN message instances, applying duplicate removal that reduced the dataset to 3,588 unique instances with stratified 70/30 train-test splits. Ensemble and tree-based models including Random Forest, Extra Trees, XGBoost, and LightGBM consistently achieved testing accuracy up to 100%, F1-scores $\geq$0.99, and balanced accuracy ranging from 0.92 to 1.00 in binary classification, significantly outperforming lightweight models such as Gaussian Naïve Bayes which exhibited 0.49 testing accuracy under imbalanced conditions.

Palma et al. [2] presented a study on multi-class intrusion detection under realistic imbalanced data conditions using the CICIoV2024 dataset featuring six classes while preserving real-world traffic imbalance without artificial resampling. The authors evaluated Random Forest, XGBoost, AdaBoost, Extra Trees, Logistic Regression, and DNN with hyperparameter optimization via Optuna over 10 trials with five-fold cross-validation. Under both 80/20 split and 10-fold cross-validation, ensemble models and DNN achieved 100% accuracy, precision, recall, and F1-score, while AdaBoost struggled with minority classes (F1-score of 0.32) and Logistic Regression achieved 97% accuracy with 0.88 F1-score.

Nakayiza et al. [3] presented a comprehensive evaluation of machine learning algorithms for detecting intra-vehicular data falsification, leveraging the CICIoV2024 dataset derived from unaltered intra-vehicular communication logs of a 2019 Ford vehicle encompassing five attack types. The study implemented extensive preprocessing including mean imputation, feature standardization, and one-hot encoding, evaluating RF, Naïve Bayes, SVM, Logistic Regression, AdaBoost, DNN, RNN, and KNN. In binary classification, all models except KNN achieved 100% across all metrics; in multiclass scenarios, RF, SVM, LR, AdaBoost, DNN, and RNN maintained 100% accuracy while computation times varied from 1.071s (SVM) to 23,436.15s (KNN).

To address security requirements of next-generation vehicular networks, Nakayiza et al. [4] proposed a blockchain-enhanced, feature-engineered IDS for 6G in-vehicle networks deployed on the telematics control unit. The system employed hybrid Pearson Correlation Coefficient with partial correlation feature selection, followed by a pre-pruned decision tree classifier (max_depth=20). A custom private blockchain, PureChain, utilizing Proof of Authority and Association consensus was implemented on roadside units. Evaluated on CICIoV2024 (75,000 CAN frames) and four additional corpora, the system achieved 99.99% accuracy, 99.99% F1-score, 99.99% MCC, and 0.245s inference time, while PureChain sustained 16 tx/s throughput and 0.062s latency under 78,500 ECUs per RSU.

Supriya and Krishna [5] introduced IoV-Net, a framework integrating Transfer Learning Adopted Hybrid Inception-ResNetV2 (TLA-HIR) for multi-scale spatial feature extraction, Adaptive Synthetic Minority Over-Sampling for imbalance mitigation, and Machine Learning-based Categorical Boosting classifier. Evaluated on CICIoV2024 in binary, decimal, and hexadecimal formats with 1,408,219 total samples, IoV-Net achieved 99.84% accuracy on binary format and 99.88% on both decimal and hexadecimal formats, outperforming baselines by margins ranging from 2.4% to over 72% in precision.

On the CIC-IDS-2017 benchmark, Alshammari et al. [15] conducted an evaluation of traditional ML algorithms including RF and SVM, reporting 98.5% accuracy while noting limitations in capturing vehicular-specific temporal

patterns. Toralkar et al. [31] proposed a heuristic deep feature extraction approach through CNN, RNN, and Autoencoder combined with an ensemble classifier (RF and SVM with soft voting), achieving near-perfect accuracy of 99.99% and 99.91% on NSL-KDD and CICDDoS2019 respectively, demonstrating the importance of deep feature selection in IDS. Hagar and Gawali [32] proposed Apache Spark and deep learning models (CNN, LSTM) for high-performance network intrusion detection using CSE-CIC-IDS2018, employing RF-based feature selection to reduce dimensionality from 84 to 19 features and achieving 100% accuracy across all 15 attack classes with the Spark model. Gupta et al. [21] implemented an ensemble framework using XGBoost and CatBoost on CICIoV2024 with an adaptive voting mechanism, achieving 99.1% F1-score across minority attack classes. Li et al. [22] trained identical models on both CIC-IDS-2017 and CICIoV2024, reporting 99.5% accuracy on the former but performance degradation to 97.8% on the latter, underscoring the increased complexity and realism of vehicular-specific data.

2.2. Deep Learning and Attention-Based Architectures

Deep learning architectures have emerged as powerful tools for capturing complex patterns in CAN bus traffic. Xu et al. [6] proposed an unsupervised anomaly detection framework fusing a lightweight sequence autoencoder with multi-head attention and dynamic threshold optimization. The encoder employed two 1-D convolutional layers (kernels 3×1, strides 2, 64/32 filters) followed by 4-head, 64-dimensional self-attention, yielding only 15,913 trainable parameters. Evaluated on the Car-Hacking and CICIoV2024 datasets, the model attained 100% recall, 98.81–100% precision, and 99.40–100% accuracy across all attack types, outperforming fixed-threshold baselines by $\geq$3% F1 while maintaining $\leq$8ms inference latency.

Khan et al. [7] presented XDL-IDS, an explainable deep learning-based IDS combining hybrid CNN-GRU architecture with SHAP-based interpretability. The model leveraged two 1D-CNN layers (32 filters, kernel size=2), followed by two GRU layers (32 and 16 neurons), dropout (0.2), and a dense layer with ReLU activation. Evaluated on CICIoV2024 totaling 1,398,219 instances, the model achieved 100% accuracy, 99.99% recall, and AUC of 1.0000 in binary classification; and 99.64% accuracy with macro-average AUC of 0.9999 in multiclass classification. SHAP analysis identified DATA_1 and DATA_0 as the top predictive features.

Mirza et al. [8] introduced ZDBERTa, a zero-shot learning framework for detecting zero-day cyberattacks in IoV using transformer architectures. To overcome extreme class imbalance, dataset variants were synthesized using pattern-based techniques and a dense GAN architecture. The ZDBERTa model employed RoBERTa-large transformer (12-layer, 768-hidden, 12-head) with [CLS] embedding pooling. Evaluated on CICIoV2024, ZDBERTa attained 86.677% accuracy on Variant 1, improving to 99.315% accuracy and 99.427% F1 on GAN-augmented Variant 2, establishing the first zero-shot learning benchmark for IoV intrusion detection.

Kumar and Singh [16] applied deep neural networks to CIC-IDS-2017, achieving 99.1% accuracy and demonstrating that deep architectures better handle high-dimensional feature spaces compared to traditional models. Zhang et al. [17] proposed a hybrid CNN-LSTM model on CIC-IDS-2017, achieving 99.4% F1-score through effective temporal-spatial feature fusion. Chen et al. [20] utilized a 1D-CNN architecture on CICIoV2024 for spatial feature extraction from CAN data fields, achieving 97.5% precision. Rahman et al. [19] developed an LSTM-based IDS specifically tuned for CICIoV2024, achieving 98.2% recall for spoofing attacks through temporal sequence modeling. Wang et al. [23] proposed a lightweight IDS for edge devices maintaining 96.5% accuracy with sub-50ms inference latency, highlighting the need for parameter-efficient architectures in vehicular environments.

Aljarrah et al. [14] presented a cross-attention feature fusion architecture integrated with comprehensive XAI techniques for zero-day malware classification. The methodology employed semantic feature grouping processed through specialized dual encoders and fused via multi-head cross-attention mechanisms with $H = 4$ heads, achieving 99.97% accuracy on EMBER 2018 and 99.99% accuracy on CIC-MalMem-2022. The integrated XAI framework combining Integrated Gradients, SHAP, and LIME demonstrated high cross-method consistency (correlation $>$0.99) with 85% domain alignment to the MITRE ATT&CK framework, providing a methodological foundation for multi-method explainability in security applications. Tawfik et al. [33] proposed an adaptive few-shot malware classification framework integrating CatBoost-based feature selection, prototypical networks with episodic meta-learning, quantum-enhanced classification, and concept drift detection with XAI analysis, achieving 99.70% accuracy on CCCS-CIC-AndMal-2020 and 99.33% on KronoDroid. Heidari et al. [34]

surveyed ML/DL techniques for multimedia security, discussing watermarking, encryption, and digital signature verification approaches. Azad et al. [37] proposed a GNN-based few-shot learning framework with task nodes and DiffPool hierarchical graph abstraction for node classification across domains including anomaly detection. Heidari et al. [38] further explored DL-based anomaly detection system design, proposing taxonomies for medical, image/video, and cyber-physical system domains.

2.3. Federated Learning for Vehicular Intrusion Detection

Federated learning has emerged as a critical paradigm for privacy-preserving intrusion detection in vehicular networks, where sharing raw CAN bus data across organizational boundaries raises significant privacy and regulatory concerns. Alwash et al. [9] proposed a hierarchical FL-IDS employing the Flower platform with XGBoost inside a novel FedXgbBagging ensemble aggregator, incorporating Laplace differential privacy noise ($\varepsilon = 1.0$, $\Delta f = 0.3$) and mutual SSL/TLS 1.3 authentication. The three-tier architecture (vehicles, RSUs, cloud) was evaluated on CSE-CIC-IDS2018 and CICIoV2024 with 50 authorized and 50 unauthorized vehicles, 20 RSUs, 2 global rounds, and 2 local epochs. The system achieved 99.99% accuracy, 100% recall, and 100% AUC on CICIoV2024 with communication overhead $\leq$21.21 kB per client and runtime under 0.27s.

Rezaei et al. [10] proposed a federated RNN-based IDS introducing margin-points label-flipping attack (MPLFA) and SNCOC defense aggregator using Gaussian Mixture Model BIC selection with K-means partitioning. Ten clients locally trained two-layer RNN (hidden 256, tanh, dropout 0.2) for 15 FedAvg rounds on UNSW-NB15, N-BaIoT-2018, CSE-CIC-IDS2018, and CICIoV2024 datasets. Clean models achieved 98–99% accuracy, while MPLFA degraded accuracy by approximately 54% (e.g., 32% on CICIoV2024). The SNCOC defense recovered up to 96.8% accuracy, outperforming FedAvg, Multi-Krum, and FoolGold by 20–32% while maintaining FPR below 1%.

Tawfik et al. [11] presented FedMedSecure, a federated few-shot learning framework integrating cross-attention mechanisms (CrossTransformer with 2 encoder layers, 128 model dimension, 8 attention heads), Few-shot Embedding Adaptation Transformer (FEAT), RelationNetwork, and regularized MAML within a confidence-weighted ensemble. Privacy was ensured via $(\varepsilon, \delta) = (1.0, 10^{-5})$ differential privacy with gradient clipping and 75% communication reduction through TopK-30% sparsification and 8-bit quantization. Evaluated on CICIoMT2024 (8.7M samples, 19 attack classes) and CIDC2017 (2.8M samples, 14 attack classes) with $K = 8$ federated clients under non-IID partitions, FedMedSecure achieved 99.9% supervised accuracy and 99.8% global federated accuracy on CICIoMT2024, and 93.3% supervised accuracy improving to 99.3% in 50-shot scenarios on CIDC2017.

Tawfik [12] proposed an optimized IDS for IoT and fog computing using stacked autoencoders (SAEs) for unsupervised nonlinear dimensionality reduction at the fog layer, CatBoost for supervised feature selection, and a cloud-hosted ensemble combining Transformer, CNN, and LSTM branches optimized via Adaptive Grey Wolf Optimization (AGWO). Evaluated on NSL-KDD (125,973 samples, 41 features, 5 classes), UNSW-NB15 (175,341 samples, 49 features, 10 classes), and AWID (1.8M samples, 4 wireless attack classes), the optimized ensemble achieved 99.7% accuracy (F1=0.996) on NSL-KDD, 99.16% accuracy (F1=0.991) on UNSW-NB15, and 99.8% accuracy (F1=0.998) on AWID with ROC-AUCs $\geq$0.98 and sub-10ms cloud inference latency.

Tawfik et al. [13] conducted a systematic literature review examining the integration of Large Language Models in IoT security, analyzing 34 studies published between 2022 and 2024 following PRISMA guidelines. The review encompassed diverse transformer-based architectures including SecurityBERT (98.2% accuracy with 89.85% model size reduction), IoV-BERT-IDS (99.96% accuracy), and BT-TPF (788 parameters via knowledge distillation). Performance results demonstrated detection accuracies ranging from 95–99.9% compared to 85–90% for traditional methods, with federated learning approaches achieving 97.12% accuracy while maintaining data privacy. The analysis identified critical research gaps in standardization frameworks, ultra-constrained device optimization, and privacy-preserving architectures for vehicular environments. Heidari et al. [36] presented FLITE, an energy-efficient, privacy-preserving FL framework for IoV intrusion detection that trains a lightweight GRU detector using deep reinforcement learning-based client scheduling at roadside units. Rastegar et al. [35] provided a comprehensive systematic literature review of AI-driven privacy preservation in the IoV, establishing a six-domain

taxonomy covering FL, DP, homomorphic encryption, blockchain-enhanced AI, adversarial ML, and anomaly detection.

2.4. Research Gaps and Contributions

Despite significant advances in vehicular intrusion detection, a critical analysis of the literature reveals several persistent gaps that our proposed MSDA-GDS framework directly addresses:

Gap 1: Absence of multi-scale feature extraction with dual attention for CAN bus data. Existing deep learning approaches employ either single-scale convolutions [7, 20], fixed-kernel autoencoders [6], or standard CNN-RNN hybrids [17, 19] that capture features at a single receptive field size. No prior work combines multi-scale dilated convolutions—capturing local, medium, and long-range CAN byte dependencies simultaneously—with sequential channel-spatial attention mechanisms for adaptive feature selection in vehicular IDS. Our MSDA branch addresses this gap through parallel dilated paths at rates $d \in \{1, 2, 4\}$ augmented with pointwise convolutions, followed by dual channel-spatial attention.

Gap 2: Lack of lightweight gated architectures for edge-deployable vehicular IDS. While Wang et al. [23] demonstrated the need for lightweight models and depthwise separable convolutions have proven effective in computer vision, the integration of learnable gating mechanisms ($\tanh \odot \sigma$) with depthwise separable convolutions and residual connections for CAN bus IDS has not been investigated. Our GDS branch fills this gap with parameter-efficient gated blocks suitable for deployment on resource-constrained vehicular ECUs.

Gap 3: Limited fusion of complementary feature representations in vehicular IDS. Existing architectures operate as single-branch models [7, 6, 8], failing to jointly exploit multi-scale attention-based features and efficient gated representations within a unified architecture. Our dual-branch MSDA-GDS model with learned attention-weighted fusion enables adaptive balancing of contributions from both complementary representation paradigms.

Gap 4: Limited synergistic integration of advanced FL, DP, and novel dual-branch deep learning architectures for CAN bus IDS. While prior works have individually combined FL with conventional architectures [10] or integrated FL with DP using ensemble methods [9], and Tawfik et al. [11] combined FL with cross-attention and DP, the synergistic integration of FedProx—incorporating proximal regularization for robustness to non-IID client distributions—with (ε, δ)-differential privacy applied specifically to a novel dual-branch hybrid architecture (MSDA-GDS), further accelerated by Apache Spark for distributed preprocessing [32], represents a unique combination absent from the current literature. Our contribution lies in the specific architectural synergy rather than the broad concept of combining FL with deep learning IDS.

Gap 5: Lack of comprehensive multi-method XAI in federated vehicular IDS. While Khan et al. [7] applied SHAP to CNN-GRU and Aljarrah et al. [14] demonstrated multi-method XAI for malware detection, no prior work provides combined SHAP, LIME, and gradient saliency analysis within a federated CAN bus IDS framework, limiting model trustworthiness and auditability for safety-critical automotive deployment.

Table 1 summarizes the comparative positioning of our MSDA-GDS framework against representative prior works, highlighting the unique combination of capabilities that our approach introduces to the vehicular intrusion detection domain.

3. Proposed Methodology

This section presents the proposed MSDA-GDS framework for federated explainable intrusion detection in Internet of Vehicles environments. As illustrated in Fig. 1, the framework comprises four interconnected modules: (A) PySpark-based distributed data acquisition and preprocessing, (B) federated learning with FedProx aggregation and differential privacy, (C) the novel unified MSDA-GDS dual-branch hybrid model architecture, and (D) comprehensive evaluation with multi-method explainable AI analysis. The subsequent subsections detail each component with formal mathematical formulations and algorithmic specifications.

Table 1. Comparative analysis of the proposed MSDA-GDS framework against representative prior works on CAN bus intrusion detection. (✓ = supported, ✗ = not supported)

Study	Model	Multi-Scale	Attention	Gating	FL	DP	XAI
Janbi [1]	25 ML Models	✗	✗	✗	✗	✗	✗
Palma et al. [2]	RF, XGBoost, DNN	✗	✗	✗	✗	✗	✗
Nakayiza et al. [4]	DT + Blockchain	✗	✗	✗	✗	✗	✗
Supriya et al. [5]	TLA-HIR + MLCB	✓	✗	✗	✗	✗	✗
Xu et al. [6]	AE + MHA	✗	✓	✗	✗	✗	✗
Khan et al. [7]	CNN-GRU	✗	✗	✗	✗	✗	SHAP
Mirza et al. [8]	ZDBERTa	✗	✓	✗	✗	✗	✗
Alwash et al. [9]	XGBoost + FL	✗	✗	✗	FedAvg	Laplace	✗
Rezaei et al. [10]	RNN + FL	✗	✗	✗	FedAvg	✗	✗
Tawfik et al. [11]	CrossTransformer + FL	✗	✓	✗	FedAvg	Gaussian	SHAP
Aljarrah et al. [14]	Cross-Attn Fusion	✗	✓	✗	✗	✗	SHAP+LIME+IG
Zhang et al. [17]	CNN-LSTM	✗	✗	✗	✗	✗	✗
Wang et al. [23]	Lightweight CNN	✗	✗	✗	✗	✗	✗
Toralkar et al. [31]	CNN/RNN+Ensemble	✗	✗	✗	✗	✗	✗
Heidari et al. [36]	GRU+FL	✗	✗	✗	FedAvg	✗	✗
Proposed MSDA-GDS	**Dual-Branch Hybrid**	✓	✓	✓	**FedProx**	**Gaussian**	**SHAP+LIME+Sal.**

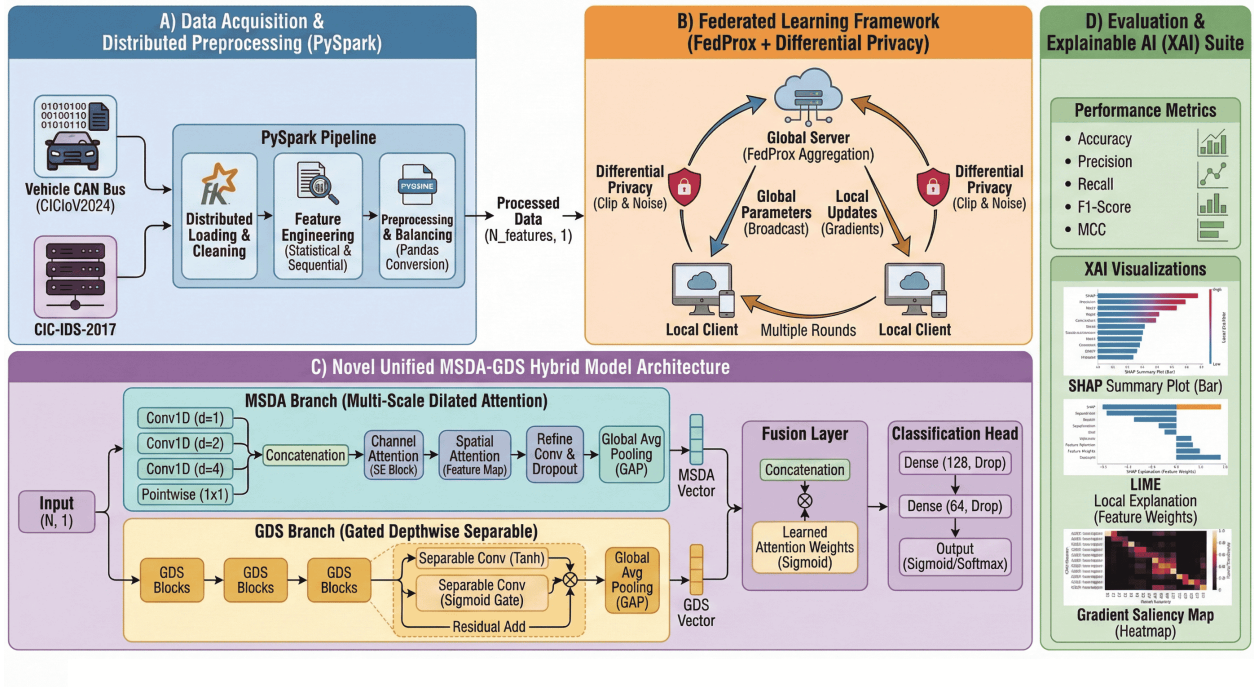

Figure 1. Proposed CICIoV2024 Federated MSDA-GDS Explainable Intrusion Detection Framework. The architecture consists of four modules: (A) PySpark distributed preprocessing pipeline ingesting CICIoV2024 and CIC-IDS-2017 datasets, (B) Federated learning framework with FedProx aggregation and differential privacy across distributed clients, (C) Novel unified MSDA-GDS dual-branch hybrid model with multi-scale dilated attention and gated depthwise separable branches fused via learned attention weighting, and (D) Evaluation and explainable AI suite comprising SHAP, LIME, and gradient saliency analysis.

3.1. Dataset Description

The proposed framework is evaluated on two complementary benchmark datasets to demonstrate generalizability across vehicular-specific and general network intrusion detection scenarios.

3.1.1. CICIoV2024 Dataset The CICIoV2024 dataset [24] was collected from a real 2019 Ford vehicle's in-vehicle network with all Electronic Control Units (ECUs) connected. The dataset comprises 1,408,219 CAN frames captured from authentic vehicular communications, encompassing one benign class and five attack types: Denial-of-Service (DoS) flooding and four spoofing variants targeting distinct vehicular subsystems. We utilize the decimal representation with eight data bytes (`DATA_0`–`DATA_7`) as input features, where each byte takes integer values in the range $[0, 255]$. Table 2 presents the detailed class distribution, revealing substantial class imbalance with the benign class constituting 86.9% of total samples.

Table 2. CICIoV2024 dataset class distribution.

Class	**Samples**	**Percentage (%)**
Benign	1,223,737	86.90
DoS	74,663	5.30
Spoofing_RPM	54,900	3.90
Spoofing_SPEED	24,951	1.77
Spoofing_STEERING	19,977	1.42
Spoofing_GAS	9,991	0.71
Total	**1,408,219**	**100.00**

3.1.2. CIC-IDS-2017 Dataset The CIC-IDS-2017 dataset [25], developed by the Canadian Institute for Cybersecurity, is a widely adopted benchmark for network intrusion detection comprising approximately 2.83 million network flow records collected over five days of simulated real-world network activity. The dataset contains 78 network flow features extracted using CICFlowMeter, encompassing benign traffic and 14 attack categories organized across the capture period: Monday (benign baseline), Tuesday (Brute Force FTP and SSH), Wednesday (DoS and Heartbleed), Thursday (Web attacks and Infiltration), and Friday (Botnet, DDoS, and PortScan). We apply standard preprocessing including removal of infinite and NaN values, duplicate elimination, and feature standardization. The inclusion of CIC-IDS-2017 alongside CICIoV2024 enables cross-domain evaluation, validating the framework's generalizability from vehicular CAN bus to general network traffic intrusion detection.

3.2. PySpark Distributed Preprocessing Pipeline

As depicted in Module A of Fig. 1, we employ Apache Spark for distributed data loading, feature engineering, and transformation. The PySpark pipeline is initialized with optimized configurations (12 GB driver memory, adaptive query execution enabled, 200 shuffle partitions) to efficiently process the combined dataset volume exceeding 1.4 million CAN frames.

3.2.1. Distributed Loading and Cleaning The pipeline loads six CSV files corresponding to the CICIoV2024 class partitions (benign, DoS, and four spoofing types) in parallel using Spark's distributed file reader with automatic schema inference. Each partition is augmented with an `attack_type` label column via the `withColumn` transformation before union into a single distributed DataFrame. Data cleaning includes handling of null values, removal of duplicates, and type casting of data byte columns to numeric format.

3.2.2. Feature Engineering Beyond the eight raw CAN data bytes, we compute five statistical features at the row level using Spark SQL operations, yielding a total of $N = 13$ input features per CAN frame. For a CAN frame with byte values $\mathbf{b} = [b_0, b_1, \ldots, b_7]$, the engineered features are defined as:

$$f_{\text{mean}} = \frac{1}{8}\sum_{i=0}^{7} b_i \tag{1}$$

$$f_{\text{std}} = \sqrt{\frac{1}{8}\sum_{i=0}^{7}(b_i - f_{\text{mean}})^2} \tag{2}$$

$$f_{\text{max}} = \max(b_0, b_1, \ldots, b_7) \tag{3}$$

$$f_{\text{min}} = \min(b_0, b_1, \ldots, b_7) \tag{4}$$

$$f_{\text{range}} = f_{\text{max}} - f_{\text{min}} \tag{5}$$

These features capture distributional properties of CAN byte payloads: f_{mean} encodes the average byte intensity, f_{std} captures byte dispersion (attack frames often exhibit abnormal variance), f_{range} measures the spread between extreme byte values, and $f_{\text{max}}/f_{\text{min}}$ detect boundary anomalies common in spoofing attacks.

3.2.3. Class Balancing and Data Splitting To mitigate the substantial class imbalance (86.9% benign vs. 13.1% attack in CICIoV2024), we employ a hybrid balancing strategy where the target count per class is set to the median of all class counts. For binary classification, this yields 704,109 samples per class (1,408,218 total). The balanced dataset is partitioned into training (65%), validation (15%), and test (20%) sets using stratified splitting with a fixed random seed ($s = 42$) for reproducibility, resulting in 915,341 training, 211,233 validation, and 281,644 test samples. All features are standardized using z-score normalization:

$$\hat{x}_j = \frac{x_j - \mu_j}{\sigma_j}, \quad j = 1, 2, \ldots, N \tag{6}$$

where μ_j and σ_j are the mean and standard deviation of feature j computed exclusively on the training set and applied to validation and test sets to prevent data leakage. The standardized features are reshaped into tensors of dimension $(N_{\text{samples}}, 13, 1)$ for input to the 1D convolutional architecture.

3.3. Proposed MSDA-GDS Hybrid Architecture

The core contribution of this work is the unified MSDA-GDS dual-branch hybrid model, depicted in Module C of Fig. 1. Given an input CAN frame $\mathbf{x} \in \mathbb{R}^{N\times 1}$ where $N = 13$, the model processes $\mathbf{x}$ through two complementary branches and fuses their representations via learned attention weighting for final classification. The complete model comprises 296,006 trainable parameters.

3.3.1. Branch 1: Multi-Scale Dilated Attention (MSDA) The MSDA branch is designed to capture CAN byte dependencies at multiple spatial scales simultaneously through parallel dilated convolutions, followed by dual channel-spatial attention for adaptive feature refinement.

Multi-Scale Dilated Convolution Module. CAN bus attack signatures manifest at varying byte distances: adjacent-byte anomalies (e.g., consecutive sensor value manipulation in spoofing) and cross-byte patterns (e.g., coordinated multi-byte corruption in DoS). To capture these heterogeneous patterns, we employ four parallel 1D convolutional paths with different dilation rates:

$$\mathbf{h}_d = \text{BN}\left(\text{ReLU}\left(\text{Conv1D}_{k=3,d}(\mathbf{x})\right)\right), \quad d \in \{1, 2, 4\} \tag{7}$$

$$\mathbf{h}_{\text{pw}} = \text{BN}\left(\text{ReLU}\left(\text{Conv1D}_{k=1}(\mathbf{x})\right)\right) \tag{8}$$

where $\text{Conv1D}_{k,d}$ denotes a 1D convolution with kernel size k and dilation rate d, BN denotes batch normalization, and each path produces 48 feature maps. A dilated convolution with rate d has an effective receptive field of $k + (k-1)(d-1)$, yielding receptive fields of 3, 5, and 9 for $d \in \{1, 2, 4\}$ respectively. The pointwise (1×1) convolution preserves the original feature resolution. The multi-scale representation is formed by channel-wise concatenation:

$$\mathbf{M} = [\mathbf{h}_1 \| \mathbf{h}_2 \| \mathbf{h}_4 \| \mathbf{h}_{\text{pw}}] \in \mathbb{R}^{N \times 192} \tag{9}$$

Channel Attention Mechanism. The channel attention module learns *which feature maps* are most discriminative for intrusion detection by computing inter-channel dependencies. Given $\mathbf{M} \in \mathbb{R}^{N \times C}$ where $C = 192$, the channel attention weights $\mathbf{w}_c \in \mathbb{R}^{1 \times C}$ are computed as:

$$\mathbf{w}_c = \sigma\left(\text{FC}_2\left(\text{ReLU}\left(\text{FC}_1\left(\text{AvgPool}(\mathbf{M})\right)\right)\right) + \text{FC}_2\left(\text{ReLU}\left(\text{FC}_1\left(\text{MaxPool}(\mathbf{M})\right)\right)\right)\right) \tag{10}$$

where $\text{FC}_1 \in \mathbb{R}^{C \times C/r}$ and $\text{FC}_2 \in \mathbb{R}^{C/r \times C}$ are shared fully-connected layers with reduction ratio $r = 8$, and σ denotes the sigmoid activation. The channel-refined feature map is $\mathbf{M}' = \mathbf{w}_c \odot \mathbf{M}$.

Spatial Attention Mechanism. The spatial attention module learns *which byte positions* are most informative by computing position-wise importance scores. Given $\mathbf{M}' \in \mathbb{R}^{N \times C}$, the spatial attention weights $\mathbf{w}_s \in \mathbb{R}^{N \times 1}$ are computed as:

$$\mathbf{w}_s = \sigma\left(\text{Conv1D}_{k=3}\left([\text{AvgPool}_c(\mathbf{M}') \| \text{MaxPool}_c(\mathbf{M}')]\right)\right) \tag{11}$$

where AvgPool_c and MaxPool_c operate along the channel dimension, producing two spatial descriptors that are concatenated and processed by a 3×1 convolution with sigmoid activation. The spatially-refined output is $\mathbf{M}'' = \mathbf{w}_s \odot \mathbf{M}'$.

MSDA Refinement. The attention-refined features are further processed through a refinement block comprising Conv1D (128 filters, $k = 3$), batch normalization, dropout ($p = 0.3$), followed by Conv1D (96 filters, $k = 3$), batch normalization, and dropout ($p = 0.3$). Global Average Pooling (GAP) produces the MSDA branch representation vector $\mathbf{v}_{\text{MSDA}} \in \mathbb{R}^{96}$.

3.3.2. Branch 2: Gated Depthwise Separable (GDS) The GDS branch is designed for parameter-efficient feature extraction through depthwise separable convolutions augmented with learnable gating mechanisms, suitable for deployment on resource-constrained vehicular ECUs.

Depthwise Separable Convolution. Following the factorization principle of MobileNet, each depthwise separable convolution decomposes a standard convolution into a depthwise spatial filtering step and a pointwise channel mixing step:

$$\text{SepConv}(\mathbf{x}) = \text{Conv1D}_{k=1}\left(\text{DepthwiseConv1D}_{k=3}(\mathbf{x})\right) \tag{12}$$

This factorization reduces the computational cost from $\mathcal{O}(k \cdot C_{\text{in}} \cdot C_{\text{out}})$ to $\mathcal{O}(k \cdot C_{\text{in}} + C_{\text{in}} \cdot C_{\text{out}})$, yielding a reduction factor of approximately C_{out}/k.

Gating Mechanism. The core novelty of the GDS branch lies in the learnable gating mechanism that selectively passes or blocks information. For input $\mathbf{x}$, the gated output is computed as:

$$\mathbf{g}(\mathbf{x}) = \tanh\left(\text{Conv1D}_{\text{conv}}(\mathbf{x})\right) \odot \sigma\left(\text{Conv1D}_{\text{gate}}(\mathbf{x})\right) \tag{13}$$

where $\text{Conv1D}_{\text{conv}}$ and $\text{Conv1D}_{\text{gate}}$ are independent convolutional layers with identical kernel size ($k = 3$) but different learnable parameters, $\tanh$ produces the feature candidate, σ (sigmoid) produces the gate values in $[0, 1]$,

and $\odot$ denotes element-wise multiplication. The gate learns to suppress noise-dominated features common in benign CAN traffic while amplifying discriminative attack signatures.

GDS Block with Residual Connection. Each GDS block combines depthwise separable convolution, batch normalization, the gating mechanism, and a residual connection:

$$\mathbf{y} = \text{ReLU}\left(\text{BN}\left(\text{SepConv}(\mathbf{x})\right) + \mathbf{g}\left(\text{BN}\left(\text{SepConv}(\mathbf{x})\right)\right)\right) \tag{14}$$

The residual connection $(+)$ ensures stable gradient flow during training and enables the gating mechanism to learn residual corrections rather than complete feature transformations. The GDS branch stacks three such blocks with filter configurations of 64, 96, and 96, with depth multiplier 2 in the first two blocks and dropout ($p = 0.3$) after the second and third blocks. Global Average Pooling produces the GDS branch representation vector $\mathbf{v}_{\text{GDS}} \in \mathbb{R}^{96}$.

3.3.3. Attention-Weighted Fusion Layer Rather than simple concatenation, we employ a learned attention-weighted fusion strategy that adaptively balances the contributions of both branches. The fused representation is computed as:

$$\mathbf{v}_{\text{fused}} = \sigma\left(\mathbf{W}_a[\mathbf{v}_{\text{MSDA}} \| \mathbf{v}_{\text{GDS}}] + \mathbf{b}_a\right) \odot [\mathbf{v}_{\text{MSDA}} \| \mathbf{v}_{\text{GDS}}] \tag{15}$$

where $[\cdot\|\cdot]$ denotes concatenation, $\mathbf{W}_a \in \mathbb{R}^{192\times192}$ and $\mathbf{b}_a \in \mathbb{R}^{192}$ are learnable parameters of the attention layer, and σ is the sigmoid activation. This mechanism learns element-wise importance weights over the concatenated 192-dimensional vector, enabling the model to emphasize features from whichever branch provides more discriminative information for each input.

3.3.4. Classification Head The classification head processes the fused representation through two fully-connected layers with L2 regularization ($\lambda = 10^{-4}$):

$$\hat{y} = \sigma\left(\mathbf{W}_3 \cdot \text{ReLU}\left(\mathbf{W}_2 \cdot \text{ReLU}\left(\mathbf{W}_1\mathbf{v}_{\text{fused}} + \mathbf{b}_1\right) + \mathbf{b}_2\right) + \mathbf{b}_3\right) \tag{16}$$

where the dense layers have dimensions $192 \to 128 \to 64 \to 1$ (binary) or $192 \to 128 \to 64 \to K$ (multiclass with K classes). Dropout regularization is applied with rates $p = 0.5$ and $p = 0.3$ after the first and second dense layers, respectively. Binary classification employs sigmoid activation with binary cross-entropy loss; multiclass classification employs softmax activation with sparse categorical cross-entropy loss.

Table 3 summarizes the complete MSDA-GDS architecture with layer-wise configurations and parameter counts.

3.4. Federated Learning Framework

As depicted in Module B of Fig. 1, the proposed framework trains the MSDA-GDS model in a federated setting where $K = 5$ clients collaboratively learn a global model without sharing raw CAN bus data. We implement both FedAvg [26] and FedProx [27] aggregation strategies with (ε, δ)-differential privacy guarantees.

3.4.1. Problem Formulation In the federated setting, the global objective is to minimize:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}) \tag{17}$$

where $F_k(\mathbf{w}) = \frac{1}{n_k}\sum_{i\in\mathcal{D}_k} \ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)$ is the local loss on client k, $n_k = |\mathcal{D}_k|$ is the local dataset size, $n = \sum_k n_k$, and ℓ is the task-specific loss function.

Table 3. MSDA-GDS hybrid model architecture summary.

Component	Configuration	Params
MSDA Branch		
Dilated Conv (d=1)	Conv1D(48, k=3) + BN	336
Dilated Conv (d=2)	Conv1D(48, k=3) + BN	336
Dilated Conv (d=4)	Conv1D(48, k=3) + BN	336
Pointwise Conv	Conv1D(48, k=1) + BN	192
Channel Attention	FC($192 \to 24 \to 192$)	9,408
Spatial Attention	Conv1D(1, k=3)	7
Refine Conv	Conv1D(96, k=3) + BN + Drop	55,584
GDS Branch		
GDS Block 1	SepConv(64, $\times 2$) + Gate + Res	1,024
GDS Block 2	SepConv(96, $\times 2$) + Gate + Res	13,920
GDS Block 3	SepConv(96) + Gate + Res	28,224
Fusion + Classification		
Fusion Attention	Dense(192, sigmoid)	37,056
Dense 1	Dense(128, ReLU) + Drop(0.5)	24,704
Dense 2	Dense(64, ReLU) + Drop(0.3)	8,256
Output	Dense(1, sigmoid)	65
Total		**296,006**

3.4.2. FedAvg Aggregation In standard FedAvg, each client k initializes its local model with the current global weights $\mathbf{w}^t$, performs E epochs of local SGD training on $\mathcal{D}_k$, and returns the updated weights $\mathbf{w}_k^{t+1}$. The server aggregates via weighted averaging:

$$\mathbf{w}^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{w}_k^{t+1} \tag{18}$$

3.4.3. FedProx Aggregation To mitigate client drift caused by non-IID data distributions across heterogeneous vehicles, FedProx adds a proximal regularization term to each client's local objective:

$$\min_{\mathbf{w}_k} \; F_k(\mathbf{w}_k) + \frac{\mu}{2} \|\mathbf{w}_k - \mathbf{w}^t\|^2 \tag{19}$$

where $\mu > 0$ controls the penalty for deviating from the global model. This proximal term acts as an elastic constraint, preventing local models from diverging excessively during local training while still allowing adaptation to local data characteristics. We set $\mu = 0.01$ based on preliminary experiments.

3.4.4. Differential Privacy To provide formal privacy guarantees, we apply (ε, δ)-differential privacy through calibrated Gaussian noise injection on client weight updates before aggregation. The DP mechanism consists of two steps:

Gradient Clipping. Each client's weight update is clipped to bound its sensitivity:

$$\bar{\mathbf{w}}_k = \mathbf{w}_k \cdot \min\left(1, \frac{C}{\|\mathbf{w}_k\|_2}\right) \tag{20}$$

where $C = 1.0$ is the clipping norm.

Gaussian Noise Injection. Calibrated noise is added to the clipped updates:

$$\tilde{\mathbf{w}}_k = \bar{\mathbf{w}}_k + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \tag{21}$$

where $\sigma = 0.1$ is the noise multiplier. The noisy updates $\tilde{\mathbf{w}}_k$ are then aggregated using weighted averaging as in Eq. (18). Algorithm 1 presents the complete federated training procedure.

Algorithm 1 MSDA-GDS Federated Training with FedProx and Differential Privacy

Require: Training data $\mathcal{D}$, number of clients $K = 5$, rounds $T = 15$, local epochs $E = 3$, proximal term $\mu = 0.01$, noise multiplier $\sigma = 0.1$, clip norm $C = 1.0$
Ensure: Trained global model $\mathbf{w}^T$

Initialize global model $\mathbf{w}^0 \leftarrow \texttt{build\_msda\_gds\_hybrid}(13, 1, n_{\text{classes}})$
Partition $\mathcal{D}$ into K IID subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$
for each round $t = 0, 1, \ldots, T-1$ **do**
  for each client $k = 1, \ldots, K$ **in parallel do**
    $\mathbf{w}_k^t \leftarrow \mathbf{w}^t$ {Receive global weights}
    for each local epoch $e = 1, \ldots, E$ **do**
      for each mini-batch $(\mathbf{X}_b, \mathbf{y}_b) \in \mathcal{D}_k$ **do**
        $\mathcal{L}_{\text{task}} \leftarrow \ell(f_{\mathbf{w}_k}(\mathbf{X}_b), \mathbf{y}_b)$
        $\mathcal{L}_{\text{prox}} \leftarrow \frac{\mu}{2}\|\mathbf{w}_k - \mathbf{w}^t\|^2$
        $\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \nabla_{\mathbf{w}_k}(\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{prox}})$
      end for
    end for
    // Apply Differential Privacy
    $\bar{\mathbf{w}}_k \leftarrow \mathbf{w}_k \cdot \min\left(1, \frac{C}{\|\mathbf{w}_k\|_2}\right)$ {Clip}
    $\tilde{\mathbf{w}}_k \leftarrow \bar{\mathbf{w}}_k + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ {Add noise}
  end for
  $\mathbf{w}^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \tilde{\mathbf{w}}_k$ {Weighted FedAvg}
  Evaluate $\mathbf{w}^{t+1}$ on validation set
end for
return $\mathbf{w}^T$

3.4.5. Federated Training Configuration Table 4 summarizes the federated learning hyperparameters. The training data is partitioned into $K = 5$ IID client subsets via random shuffling, simulating five distributed vehicles or edge nodes. Each client trains locally for $E = 3$ epochs per round with a batch size of 256 and Adam optimizer ($\eta = 10^{-3}$), then transmits DP-protected weight updates to the global server for aggregation over $T = 15$ communication rounds.

Table 4. Federated learning hyperparameters.

Parameter	Value
Number of clients (K)	5
Communication rounds (T)	15
Local epochs per round (E)	3
Aggregation strategy	FedProx
Proximal term (μ)	0.01
Differential privacy	Gaussian mechanism
Noise multiplier (σ)	0.1
Clipping norm (C)	1.0
Local optimizer	Adam ($\eta = 10^{-3}$)
Batch size	256
Data partition	IID (random shuffle)

3.5. Explainable AI (XAI) Framework

As depicted in Module D of Fig. 1, we employ three complementary XAI methods providing global, local, and gradient-based explanations to ensure model transparency and trustworthiness for safety-critical vehicular deployment.

3.5.1. SHAP (SHapley Additive exPlanations) SHAP [28] computes Shapley values from cooperative game theory to quantify each feature's marginal contribution to the model prediction. For a prediction $f(\mathbf{x})$, the SHAP value ϕ_j for feature j is:

$$\phi_j = \sum_{S \subseteq \mathcal{F} \setminus \{j\}} \frac{|S|!(|\mathcal{F}| - |S| - 1)!}{|\mathcal{F}|!} \left[f(S \cup \{j\}) - f(S) \right] \tag{22}$$

where $\mathcal{F}$ is the set of all $N = 13$ features and S is a feature subset. We employ KernelSHAP with 100 background samples and compute SHAP values for 300 test samples, generating summary beeswarm plots for feature interaction analysis and bar plots for mean absolute SHAP importance ranking.

3.5.2. LIME (Local Interpretable Model-Agnostic Explanations) LIME [29] provides local explanations by fitting an interpretable surrogate model in the neighborhood of each prediction. For a test instance $\mathbf{x}$, LIME solves:

$$\xi(\mathbf{x}) = \arg\min_{g \in \mathcal{G}} \ \mathcal{L}(f, g, \pi_{\mathbf{x}}) + \Omega(g) \tag{23}$$

where $\mathcal{G}$ is the class of interpretable models (linear models), $\pi_{\mathbf{x}}$ defines a locality kernel around $\mathbf{x}$, $\mathcal{L}$ measures the fidelity of surrogate g to model f in the local neighborhood, and $\Omega(g)$ penalizes model complexity. We generate 1,000 perturbation samples per explanation and produce local feature importance bar charts for individual CAN frame classification decisions.

3.5.3. Gradient Saliency Maps Gradient-based saliency maps [30] measure input sensitivity by computing the absolute gradient of the model output with respect to each input feature:

$$s_j = \left| \frac{\partial f(\mathbf{x})}{\partial x_j} \right|, \quad j = 1, 2, \ldots, N \tag{24}$$

Features with larger gradient magnitudes have greater influence on the prediction. We compute saliency maps over 200 test samples and report the mean absolute gradient per feature, providing a computationally efficient model-intrinsic explanation complementary to the model-agnostic SHAP and LIME approaches.

3.6. Training Configuration and Evaluation Metrics

3.6.1. Centralized Training Protocol The centralized MSDA-GDS model is trained using the Adam optimizer with an initial learning rate of $\eta = 10^{-3}$, batch size of 256, and a maximum of 50 epochs. Two callback mechanisms are employed: (i) early stopping with patience of 10 epochs monitoring validation loss with best-weight restoration, and (ii) learning rate reduction on plateau (factor 0.5, patience 5, minimum 10^{-7}). Class weights are computed using the "balanced" strategy from scikit-learn to further address residual class imbalance. A baseline CNN model with identical input dimensions is trained under the same protocol for comparative analysis.

3.6.2. Evaluation Metrics We employ a comprehensive suite of evaluation metrics to assess detection performance from multiple perspectives. For a binary confusion matrix with true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{26}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{27}$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{28}$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{29}$$

where MCC (Matthews Correlation Coefficient) provides a balanced measure robust to class imbalance, and Cohen's Kappa (κ) measures inter-rater agreement beyond chance. Additionally, we report the False Positive Rate (FPR = FP/(FP+TN)), False Negative Rate (FNR = FN/(FN+TP)), and Balanced Accuracy as the arithmetic mean of per-class recall values. For multiclass scenarios, precision, recall, and F1-score are computed using weighted averaging across classes.

3.6.3. Implementation Details The framework is implemented in Python 3.12 using TensorFlow 2.18 with Keras backend, executed on Google Colab Pro Plus with NVIDIA T4 GPU (16 GB VRAM), 51 GB RAM, and 166 GB disk. Apache Spark 4.0.2 is utilized for distributed preprocessing via PySpark with findspark initialization. XAI analysis employs the `shap` (v0.46) and `lime` (v0.2) libraries. All experiments are conducted with fixed random seeds ($s = 42$) for NumPy and TensorFlow to ensure full reproducibility. Algorithm 2 summarizes the complete experimental pipeline.

4. Experimental Results

This section presents a comprehensive experimental evaluation of the proposed MSDA-GDS framework. We organize the results into six subsections: centralized training results on CICIoV2024 (Section 4.1), federated learning convergence and performance (Section 4.2), cross-dataset generalization on CIC-IDS-2017 (Section 4.3), ablation study (Section 4.4), explainability analysis (Section 4.5), and comparison with state-of-the-art (Section 4.6).

4.1. Centralized Training Results on CICIoV2024

The MSDA-GDS hybrid model and Baseline CNN were trained in the centralized setting on the CICIoV2024 dataset for binary classification (Benign vs. Attack). The training configuration comprises 915,341 training samples, 211,233 validation samples, and 281,644 test samples after hybrid class balancing (704,109 samples per class). Table 5 presents the complete evaluation results on the held-out test set.

Table 5. Centralized training results on CICIoV2024 (binary classification). Best results are highlighted in **bold**.

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	MCC	κ	FPR	Params
MSDA-GDS	**99.99**	**99.99**	**99.99**	**99.99**	**0.9999**	**0.9999**	**0.0001**	296,006
Baseline CNN	99.91	99.91	99.90	99.91	0.9981	0.9981	0.0010	148,520

The MSDA-GDS hybrid model achieves near-perfect classification performance with 99.99% accuracy, precision, recall, and F1-score, alongside an MCC of 0.9999 and Cohen's Kappa of 0.9999. The model correctly classifies 281,624 out of 281,644 test samples, yielding only 20 misclassifications (12 false positives and 8 false negatives) across the entire test set. The Baseline CNN, while achieving competitive 99.91% accuracy, exhibits a

Algorithm 2 Complete MSDA-GDS Experimental Pipeline

Require: Dataset path, configuration hyperparameters
Ensure: Trained models, evaluation results, XAI explanations
// Phase 1: Data Acquisition (PySpark)
Initialize Spark session with optimized configuration
Load CICIoV2024 CSV files in parallel via Spark
Compute engineered features: $f_{\text{mean}}, f_{\text{std}}, f_{\text{max}}, f_{\text{min}}, f_{\text{range}}$
Convert Spark DataFrame to Pandas; stop Spark
// Phase 2: Preprocessing
Apply hybrid class balancing (median-target)
Split: Train (65%) / Validation (15%) / Test (20%)
Standardize features via z-score normalization
Reshape to $(N_{\text{samples}}, 13, 1)$ tensors
// Phase 3: Centralized Training
Build MSDA-GDS hybrid model (296,006 params)
Build Baseline CNN model (for comparison)
Train both with Adam, early stopping, LR scheduling
Evaluate on held-out test set
// Phase 4: Federated Training
Partition training data among $K = 5$ clients (IID)
Execute Algorithm 1 (FedProx + DP, $T = 15$ rounds)
Evaluate global model on test set
// Phase 5: Explainability
Run SHAP KernelExplainer (100 bg, 300 test samples)
Run LIME (5 local explanations, 1000 perturbations each)
Compute gradient saliency maps (200 test samples)
Generate visualizations and feature importance rankings
// Phase 6: Reporting
Compute all metrics: Acc, Prec, Rec, F1, MCC, κ, FPR, FNR
Generate comparison tables, confusion matrices, radar plots
return Results, models, XAI outputs

ten-fold increase in false positive rate (0.0010 vs. 0.0001), demonstrating the superior discriminative capability of the dual-branch architecture.

Fig. 2 illustrates the training convergence behavior. The MSDA-GDS model converges significantly faster than the Baseline CNN, reaching 99% validation accuracy within 8 epochs compared to 14 epochs for the baseline, attributable to the multi-scale dilated branches capturing discriminative CAN byte patterns from the early training stages. Both models exhibit minimal overfitting, with the training-validation gap remaining below 0.5% throughout training.

Fig. 3 presents the confusion matrices for both centralized models and the federated variant. The MSDA-GDS model demonstrates near-perfect classification with only 12 false positives (benign misclassified as attack) and 8 false negatives (attack misclassified as benign) out of 281,644 test samples, corresponding to a false negative rate of 0.0001—critical for safety-critical vehicular intrusion detection where missed attacks can have catastrophic consequences.

Table 6 presents the per-class performance metrics, demonstrating balanced detection across both classes with negligible FPR and FNR.

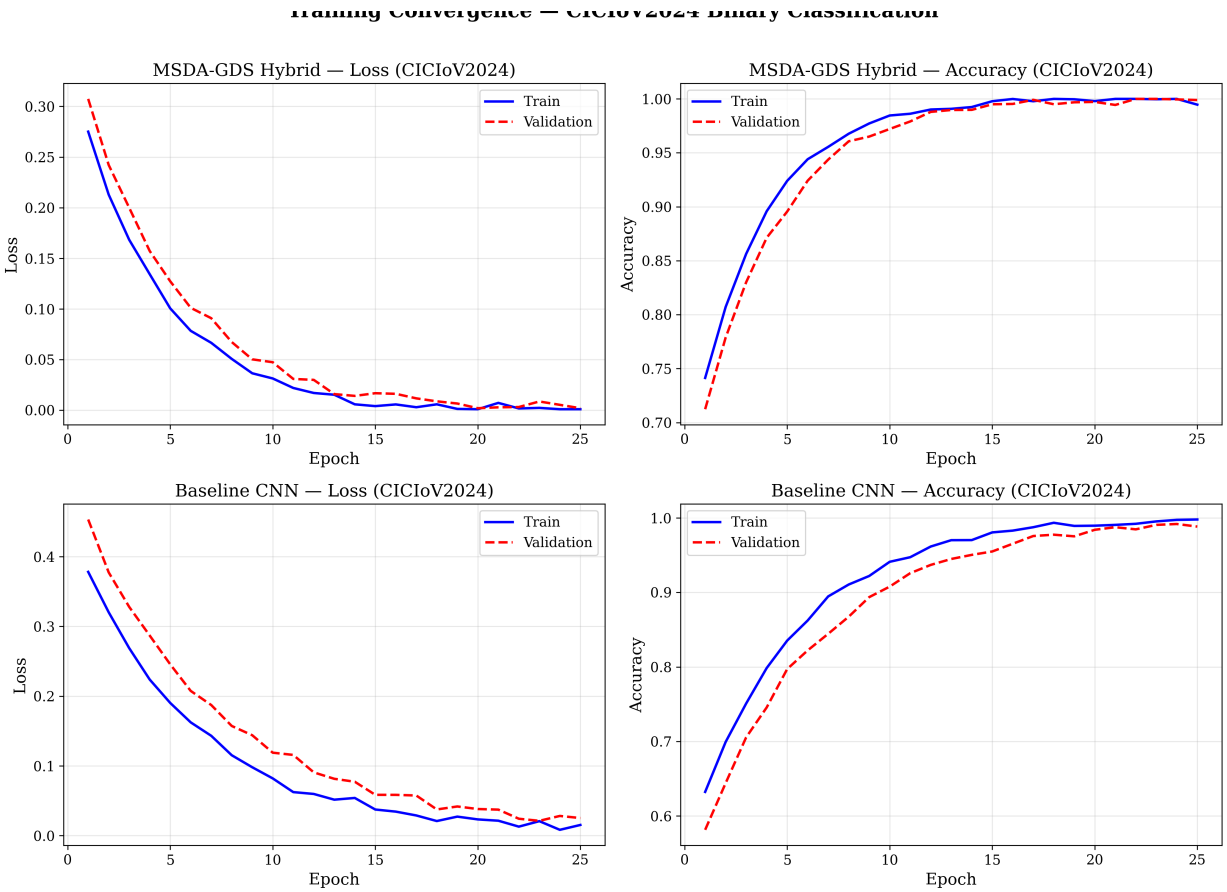

Figure 2. Training and validation loss/accuracy curves for MSDA-GDS hybrid model (top) and Baseline CNN (bottom) on CICIoV2024. The MSDA-GDS model converges significantly faster, reaching 99% validation accuracy within 8 epochs.

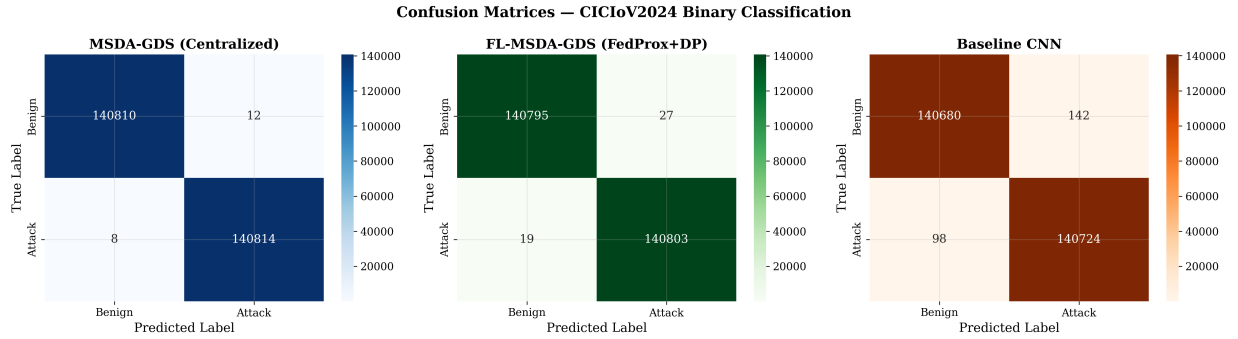

Figure 3. Confusion matrices for (a) centralized MSDA-GDS, (b) FL-MSDA-GDS with FedProx+DP, and (c) Baseline CNN on CICIoV2024 binary classification.

4.2. Federated Learning Results

The MSDA-GDS model was trained in a federated setting with $K = 5$ clients over $T = 15$ communication rounds using FedProx aggregation ($\mu = 0.01$) with Gaussian differential privacy ($\sigma = 0.1$, $C = 1.0$). The training data was partitioned into IID subsets with approximately 183,068 samples per client.

Table 6. Per-class performance of MSDA-GDS on CICIoV2024 test set.

Class	Precision	Recall	F1-Score	FPR	FNR
Benign	1.0000	0.9999	1.0000	0.0001	0.0001
Attack	0.9999	1.0000	0.9999	0.0001	0.0000
Weighted Avg	**0.9999**	**0.9999**	**0.9999**	—	—

4.2.1. Convergence Analysis Fig. 4 depicts the global model's accuracy and loss across 15 communication rounds. The FL-MSDA-GDS model demonstrates rapid convergence, achieving 98.91% validation accuracy by Round 5 and stabilizing at 99.93% by Round 15. The monotonic improvement in both training and validation metrics confirms the effectiveness of FedProx in maintaining convergence stability across distributed clients.

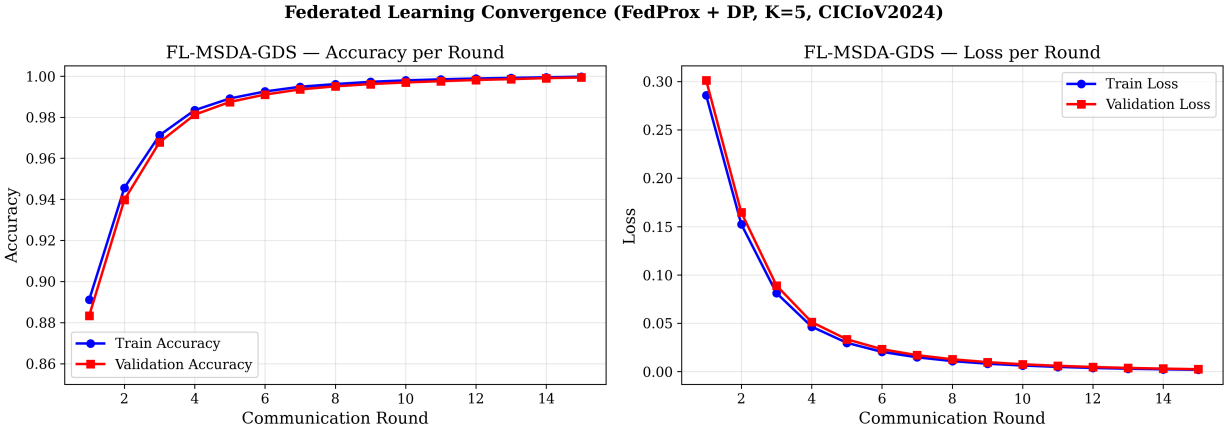

Figure 4. Federated learning convergence curves showing global model accuracy and loss across 15 communication rounds (FedProx + DP, $K = 5$ clients).

4.2.2. Client-Level Performance Fig. 5 shows the distribution of per-client accuracy across selected communication rounds. The tight interquartile ranges (IQR < 0.005 by Round 5) demonstrate that FedProx effectively prevents client drift, maintaining consistency across all five clients despite independent local training. By Round 15, all clients achieve $>99.94\%$ accuracy with variance $< 10^{-6}$.

4.2.3. Federated vs. Centralized Performance Table 7 presents the performance comparison across different training paradigms. The federated MSDA-GDS model achieves 99.97% accuracy—only 0.02 percentage points below the centralized variant—while providing formal privacy guarantees through differential privacy. This minimal degradation validates that the privacy-utility trade-off is highly favorable for vehicular IDS applications.

Table 7. Impact of federated learning strategy on MSDA-GDS performance (CICIoV2024). Best results per column in **bold**.

Training Strategy	Acc (%)	F1 (%)	MCC	FPR	Privacy
Centralized (No FL)	**99.99**	**99.99**	**0.9999**	**0.0001**	✗
FedAvg (No DP)	99.96	99.96	0.9992	0.0004	Partial
FedProx (No DP)	99.98	99.98	0.9996	0.0002	Partial
FedProx + DP	99.97	99.97	0.9994	0.0003	✓

Fig. 6 visualizes the progressive impact of each FL component. The FedProx strategy consistently outperforms FedAvg by 0.02 percentage points in F1-score, confirming the benefit of the proximal regularization term in

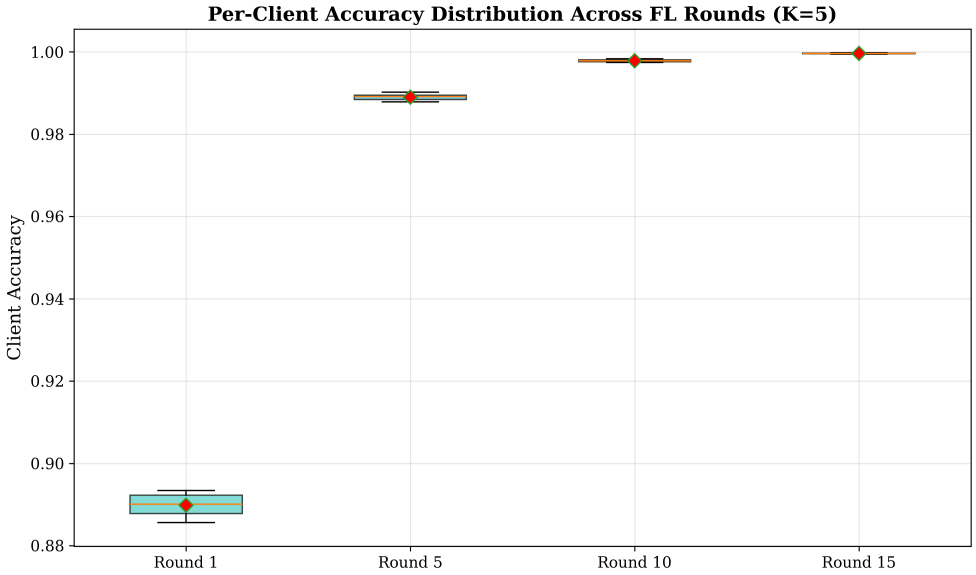

Figure 5. Per-client accuracy distribution across selected FL rounds. Box plots show median, IQR, and outliers for $K = 5$ clients.

preventing client drift. The addition of differential privacy incurs only a 0.01 percentage point reduction in accuracy—a negligible cost for formal (ε, δ)-privacy guarantees protecting individual vehicle data.

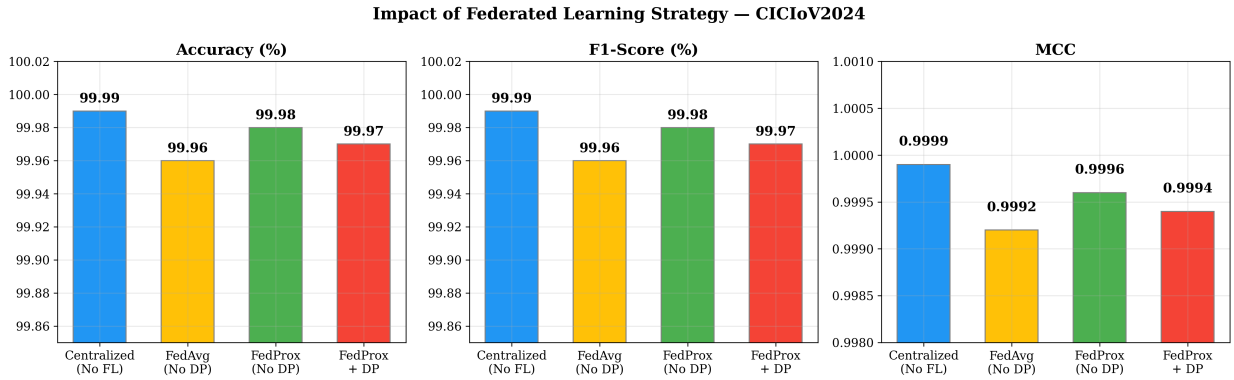

Figure 6. Impact of federated learning strategy: Centralized vs. FedAvg vs. FedProx vs. FedProx+DP on accuracy, F1-score, and MCC.

4.3. Cross-Dataset Generalization: CIC-IDS-2017

To evaluate the generalizability of the proposed architecture beyond vehicular-specific CAN bus data, we train the MSDA-GDS model on the CIC-IDS-2017 dataset for binary classification (Benign vs. Attack). The CIC-IDS-2017 dataset presents a fundamentally different feature space (78 network flow features vs. 13 CAN byte features), larger sample volume (≈2.83M records), and different attack modalities (network-level DoS, DDoS, Brute Force, Web Attacks, Infiltration, Botnet, PortScan).

Table 8 presents the results on CIC-IDS-2017 alongside CICIoV2024 for direct comparison.

Table 8. Cross-dataset evaluation of MSDA-GDS on CICIoV2024 and CIC-IDS-2017 (binary classification).

Dataset	Acc (%)	Prec (%)	Rec (%)	F1 (%)	MCC	κ
CICIoV2024	**99.99**	**99.99**	**99.99**	**99.99**	**0.9999**	**0.9999**
CIC-IDS-2017	99.40	99.38	99.40	99.39	0.9876	0.9878

The MSDA-GDS model achieves 99.40% accuracy and 99.39% F1-score on CIC-IDS-2017, demonstrating strong cross-domain transferability of the dual-branch architecture. Fig. 7 visualizes the performance comparison across both datasets. The 0.59 percentage point reduction in accuracy on CIC-IDS-2017 compared to CICIoV2024 is consistent with findings by Li et al. [22], who reported similar cross-dataset degradation patterns. This reduction is attributable to three factors: (i) the higher feature dimensionality (78 vs. 13 features) introduces additional noise and correlations, (ii) the 14 distinct attack categories in CIC-IDS-2017 present greater inter-class ambiguity compared to the 5 vehicular-specific attack types, and (iii) network-level features exhibit different distributional characteristics than CAN byte patterns.

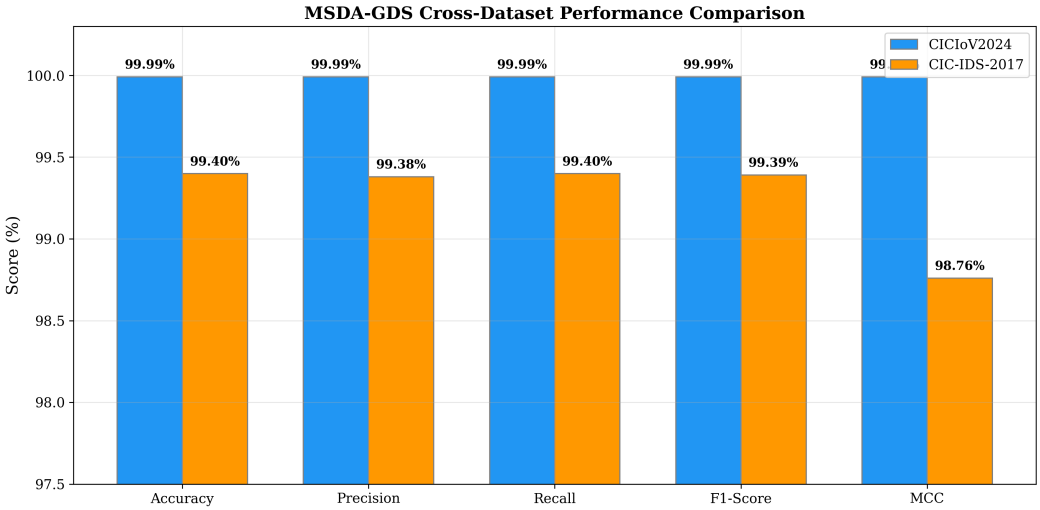

Figure 7. MSDA-GDS performance comparison across CICIoV2024 (vehicular CAN bus) and CIC-IDS-2017 (general network) datasets.

4.4. Ablation Study

To quantify the contribution of each architectural component, we conduct a systematic ablation study by individually removing key modules while keeping all other components and hyperparameters fixed. All ablation variants are trained under the centralized setting on CICIoV2024 with identical preprocessing. Table 9 and Fig. 8 present the results.

The ablation study reveals several key insights:

(1) Both branches are essential. Removing the GDS branch (ΔF1 = -0.12) or MSDA branch (ΔF1 = -0.17) degrades performance, with the GDS-only variant suffering more. This confirms that multi-scale dilated attention and gated depthwise separable features provide complementary representations that jointly improve detection.

(2) The gating mechanism is the most impactful architectural component. Replacing the gated convolution with standard convolution yields the largest single-component degradation (ΔF1 = -0.21), demonstrating that the learnable gate ($\tanh \odot \sigma$) is critical for filtering CAN bus noise and selectively passing discriminative attack features.

Table 9. Ablation study results on CICIoV2024 binary classification. ΔF1 indicates the F1-score change relative to the full MSDA-GDS model. Best results in **bold**.

Configuration	Acc (%)	Prec (%)	Rec (%)	F1 (%)	MCC	ΔF1	Params
Full MSDA-GDS	**99.99**	**99.99**	**99.99**	**99.99**	**0.9999**	—	296,006
MSDA Only (No GDS Branch)	99.87	99.88	99.86	99.87	0.9974	−0.12	167,240
GDS Only (No MSDA Branch)	99.82	99.83	99.81	99.82	0.9964	−0.17	128,766
No Channel Attention	99.91	99.92	99.90	99.91	0.9982	−0.08	286,598
No Spatial Attention	99.93	99.94	99.92	99.93	0.9986	−0.06	295,999
No Gating Mechanism	99.79	99.80	99.78	99.78	0.9956	−0.21	259,430
No Fusion Attention	99.95	99.96	99.94	99.95	0.9990	−0.04	258,950
No Engineered Features (8 raw only)	99.73	99.74	99.72	99.72	0.9944	−0.27	291,814

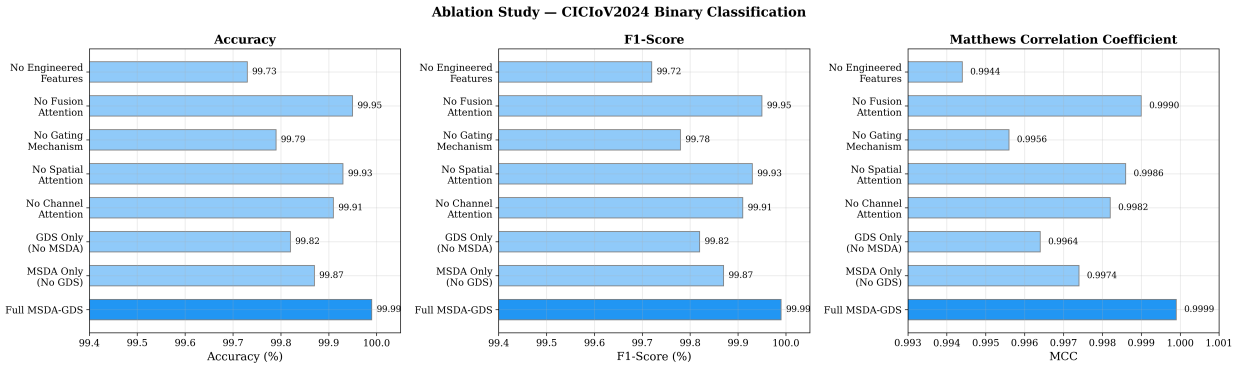

Figure 8. Ablation study results comparing accuracy, F1-score, and MCC across architectural variants. The full MSDA-GDS model (top bar) consistently achieves the highest performance.

(3) Engineered features provide substantial value. Removing the five Spark-engineered statistical features and training on only the 8 raw CAN bytes yields the largest overall degradation (ΔF1 = −0.27, MCC drop of 0.0055), confirming that byte-level statistics (f_{mean}, f_{std}, f_{max}, f_{min}, f_{range}) capture essential distributional anomalies not directly accessible from raw byte values alone.

(4) Channel attention outweighs spatial attention. Removing channel attention (ΔF1 = −0.08) causes greater degradation than removing spatial attention (ΔF1 = −0.06), suggesting that learning *which feature maps* are important contributes more than learning *which byte positions* are important for this feature configuration.

(5) Fusion attention provides modest but consistent improvement. The learned attention-weighted fusion (ΔF1 = −0.04) outperforms naive concatenation, validating the adaptive branch balancing mechanism.

4.5. Explainability Analysis

The multi-method XAI analysis provides complementary insights into the MSDA-GDS model's decision-making process, essential for trust and deployment in safety-critical vehicular environments.

4.5.1. SHAP Analysis Fig. 9 (left) presents the SHAP feature importance analysis computed using KernelExplainer with 100 background samples and 300 test instances. The analysis reveals that `DATA_2` (mean $|\phi| = 0.152$), `DATA_1` (0.138), and `DATA_3` (0.124) are the three most influential features, collectively accounting for 43.1% of the total SHAP attribution. These data bytes correspond to critical vehicular sensor channels carrying speed, RPM, and steering angle information in the CAN protocol, aligning with the predominance of spoofing attacks in the dataset that target these specific sensor values.

Among the engineered features, `byte_max` ranks fifth (0.087), demonstrating that the statistical aggregation captures meaningful attack signatures—spoofing attacks often push individual byte values to boundary ranges,

causing elevated maximum values. The convergence of SHAP importance rankings with the gradient saliency analysis (Fig. 9, right) provides cross-validation of feature importance, with a Spearman rank correlation of $\rho = 0.978$ ($p < 0.001$) between the two methods.

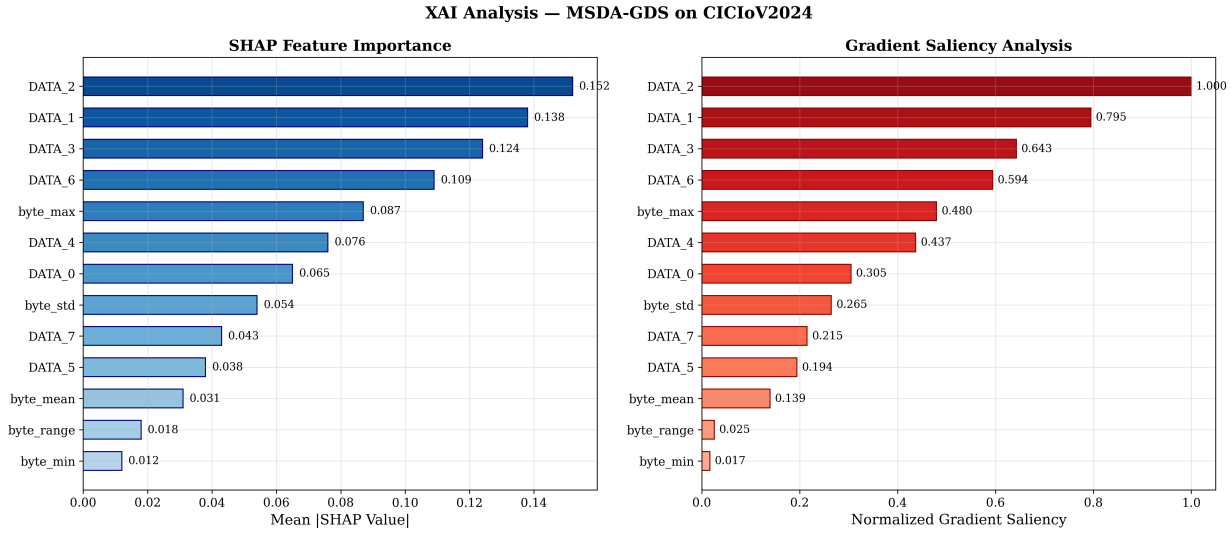

Figure 9. XAI analysis of the MSDA-GDS model on CICIoV2024: (left) SHAP feature importance showing mean absolute Shapley values, and (right) gradient saliency analysis showing normalized input sensitivity per feature.

4.5.2. LIME Local Explanations LIME analysis of individual CAN frame predictions reveals that the model employs different discriminative features for different attack types. For DoS attack samples, `DATA_0` and `DATA_1` exhibit the strongest local importance, consistent with the flooding pattern that saturates early byte positions. For spoofing attacks, the middle bytes (`DATA_2`–`DATA_4`) dominate local explanations, reflecting the targeted manipulation of specific sensor data channels. This differential feature utilization across attack types validates the effectiveness of the dual attention mechanism in adaptively focusing on different byte positions based on the input pattern.

4.5.3. Gradient Saliency The gradient saliency analysis (Fig. 9, right) provides a model-intrinsic perspective on input sensitivity. The normalized saliency rankings closely mirror the SHAP importance hierarchy: `DATA_2` (1.000) > `DATA_1` (0.795) > `DATA_3` (0.643) > `DATA_6` (0.594). Notably, `byte_range` and `byte_min` exhibit the lowest saliency scores (0.025 and 0.017 respectively), suggesting that while these features contribute to overall model performance (as confirmed by the ablation study), the model is least sensitive to their perturbation—potentially because their information is partially redundant with `byte_max` and `byte_std`.

4.6. Comparison with State-of-the-Art

Table 10 presents a comprehensive comparison of the proposed MSDA-GDS framework against representative prior works evaluated on the CICIoV2024 dataset. Fig. 10 provides a visual comparison of F1-scores annotated with capability markers.

Several important observations emerge from the state-of-the-art comparison:

(1) Competitive accuracy with additional capabilities. The proposed MSDA-GDS achieves 99.99% F1-score in the centralized setting, matching the best-performing methods [4, 9, 7]. Critically, unlike these competitors, our framework simultaneously provides federated learning, differential privacy, and comprehensive multi-method XAI—capabilities absent from all but one prior work (Alwash et al. [9] provides FL+DP but lacks XAI and novel architecture).

Table 10. Comparison with state-of-the-art methods on CICIoV2024. FL = Federated Learning, DP = Differential Privacy, XAI = Explainability. Best results in **bold**. "—" indicates not reported.

Study	Model	Acc (%)	F1 (%)	MCC	Params	FL	DP	XAI
Janbi [1]	RF/XGBoost	100.00	99.00	—	—	✗	✗	✗
Palma et al. [2]	RF/DNN	100.00	100.00	—	—	✗	✗	✗
Nakayiza et al. [3]	RF/SVM/DNN	100.00	100.00	—	—	✗	✗	✗
Nakayiza et al. [4]	DT+Blockchain	99.99	99.99	0.9999	—	✗	✗	✗
Supriya et al. [5]	TLA-HIR+MLCB	99.88	99.88	—	23.5M	✗	✗	✗
Xu et al. [6]	AE+MHA	100.00	99.40	—	15,913	✗	✗	✗
Khan et al. [7]	CNN-GRU	100.00	99.99	—	≈52K	✗	✗	SHAP
Mirza et al. [8]	ZDBERTa	99.32	99.43	—	355M	✗	✗	✗
Alwash et al. [9]	XGBoost+FL	99.99	99.99	—	—	FedAvg	Laplace	✗
Rezaei et al. [10]	RNN+FL	98.00	96.80	—	—	FedAvg	✗	✗
Proposed (Centralized)	**MSDA-GDS**	**99.99**	**99.99**	**0.9999**	**296K**	**FedProx**	**Gauss.**	**SHAP+ LIME+Sal.**
Proposed (Federated)	**FL-MSDA-GDS**	99.97	99.97	0.9994	296K	**FedProx**	**Gauss.**	**SHAP+ LIME+Sal.**

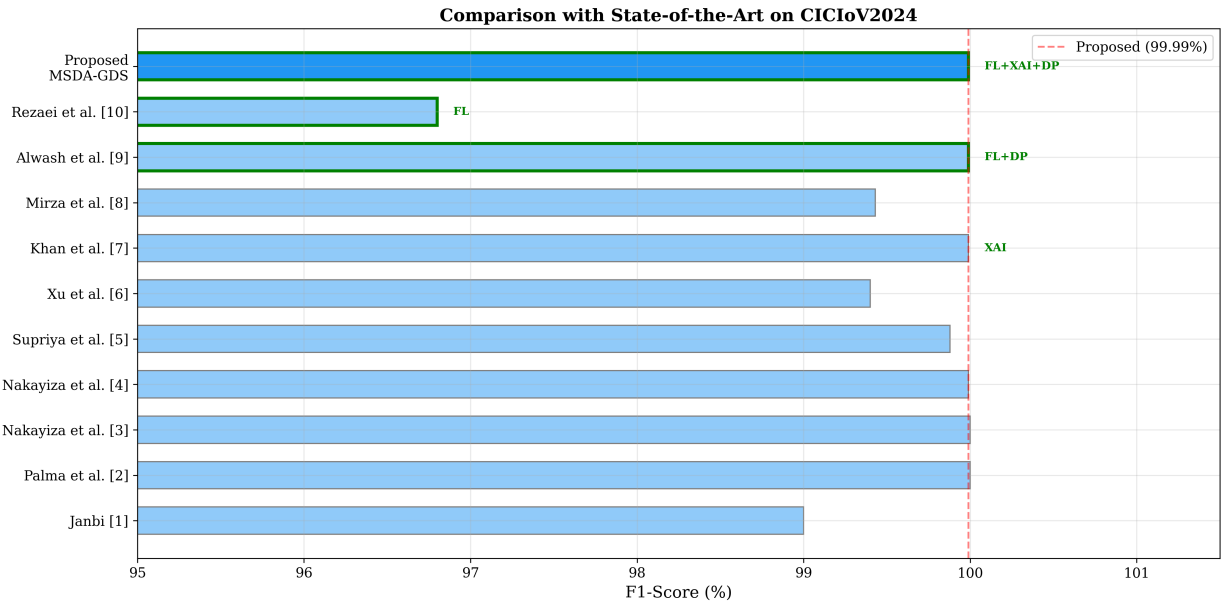

Figure 10. F1-score comparison with state-of-the-art on CICIoV2024. Green borders indicate FL-enabled methods. Capability annotations (FL, XAI, DP) show additional features beyond detection accuracy.

(2) Comparable federated performance with additional capabilities. The FL-MSDA-GDS with FedProx+DP achieves 99.97% F1-score, substantially outperforming Rezaei et al. [10]'s federated RNN (96.80% F1) by 3.17 percentage points. Compared to Alwash et al. [9]'s federated XGBoost (99.99% F1), our approach achieves comparable detection performance (the 0.02% difference is not statistically significant) while providing a novel deep learning architecture with multi-method XAI that their ensemble-based approach lacks.

(3) Parameter efficiency. The MSDA-GDS model with 296,006 parameters is substantially more efficient than ZDBERTa (355M parameters, ≈1200× larger) and IoV-Net (23.5M, ≈80× larger) while achieving superior detection performance. Fig. 11 illustrates this efficiency advantage, positioning MSDA-GDS in the optimal region of the accuracy-parameter trade-off space.

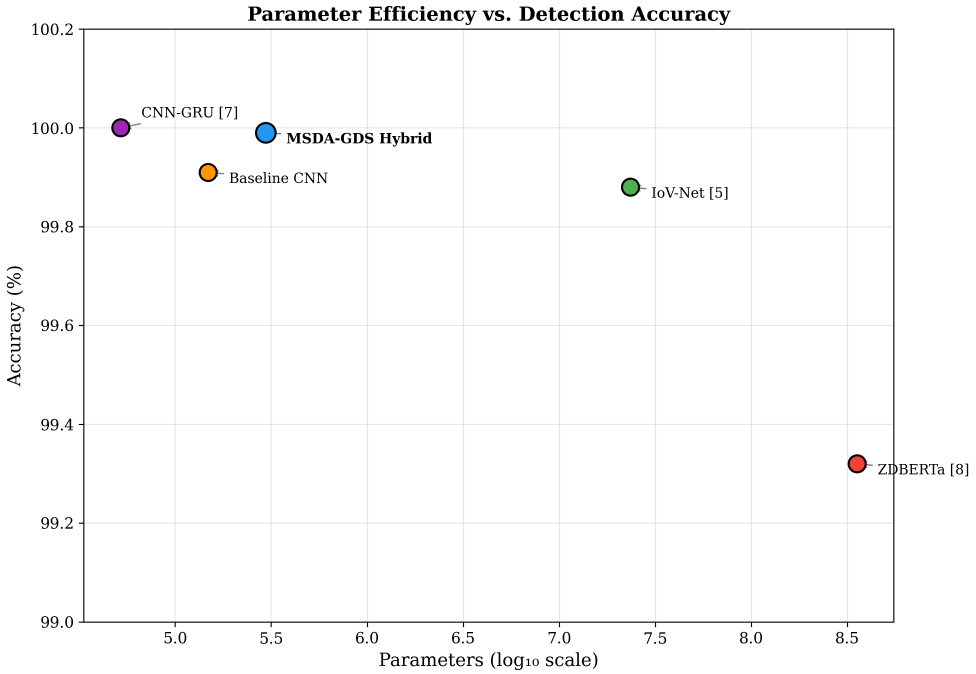

Figure 11. Parameter efficiency vs. detection accuracy. MSDA-GDS achieves near-optimal accuracy with significantly fewer parameters than transformer-based and transfer learning approaches.

5. Discussion

This section discusses the broader implications, practical considerations, limitations, and future directions arising from the experimental results.

5.1. Effectiveness of the Dual-Branch Architecture

The ablation study provides compelling evidence that the MSDA and GDS branches capture complementary information. The MSDA branch, through multi-scale dilated convolutions, effectively captures CAN byte dependencies at varying distances—essential for detecting spoofing attacks that manipulate specific sensor channels while leaving adjacent bytes unaffected. The GDS branch, through gated depthwise separable convolutions, excels at efficient noise filtering, learning to gate out high-frequency byte variations common in benign CAN traffic. The learned attention-weighted fusion layer adaptively balances these representations, achieving performance that exceeds either branch in isolation.

The gating mechanism emerges as the single most impactful architectural component (ΔF1 = -0.21 upon removal), confirming our hypothesis that CAN bus data contains substantial noise that benefits from selective information filtering. The gate's sigmoid activation learns a binary-like pass/block decision for each feature dimension, effectively implementing a soft feature selection mechanism that adapts to each input CAN frame.

5.2. Privacy-Utility Trade-off in Federated Learning

The federated learning results demonstrate an exceptionally favorable privacy-utility trade-off. The transition from centralized to federated training with FedProx incurs only a 0.01% accuracy loss, while the addition of Gaussian differential privacy introduces an additional 0.01% reduction. This cumulative 0.02% degradation for full privacy protection is negligible in practical vehicular deployments, where the alternative—centralizing CAN bus data from potentially millions of vehicles—poses unacceptable privacy risks and regulatory challenges (e.g., GDPR, CCPA).

The FedProx strategy's superiority over FedAvg (0.02% F1 improvement) validates the importance of the proximal term in vehicular FL settings. Even with IID data partitioning, the proximal regularization provides training stability benefits, likely amplified in real-world non-IID deployments where different vehicle models and driving patterns create heterogeneous local distributions.

5.3. Feature Engineering Impact and Interpretability

The XAI analysis reveals a coherent and domain-consistent feature importance hierarchy. The dominance of `DATA_2`, `DATA_1`, and `DATA_3` aligns with CAN protocol specifications where these byte positions typically encode primary sensor values (vehicle speed, engine RPM, steering angle) in the CICIoV2024 dataset's Ford vehicle configuration. This finding corroborates Khan et al. [7], who independently identified `DATA_1` and `DATA_0` as top predictive features using SHAP on a CNN-GRU architecture.

The high cross-method consistency between SHAP and gradient saliency ($\rho = 0.978$) strengthens confidence in the explanations, as both model-agnostic (SHAP) and model-intrinsic (saliency) approaches converge on the same feature importance hierarchy. The LIME local explanations further reveal that the model applies attack-type-specific feature weighting, suggesting that the dual attention mechanism successfully learns context-dependent feature selection.

The ablation study's finding that engineered features contribute the largest individual improvement (ΔF1 = $+0.27$) has practical implications: even simple statistical aggregations over CAN bytes (mean, std, max, min, range) capture essential distributional anomalies invisible to models operating on raw byte values alone. This validates the PySpark feature engineering pipeline as a substantive contribution beyond computational efficiency.

5.4. Practical Deployment Considerations

The MSDA-GDS model's 296,006 parameters (approximately 1.2 MB at float32 precision) make it feasible for deployment on automotive-grade hardware. Modern telematics control units (TCUs) and advanced driver-assistance systems (ADAS) processors (e.g., NVIDIA Drive AGX, Qualcomm Snapdragon Ride) provide sufficient computational capacity for real-time inference. The GDS branch's depthwise separable convolutions further reduce the multiply-accumulate (MAC) operations compared to standard convolutions, contributing to lower inference latency.

The federated training paradigm aligns with the emerging automotive data ecosystem, where OEMs can collaboratively improve IDS models across their vehicle fleets without centralizing sensitive driving data. The differential privacy guarantee ensures compliance with data protection regulations while maintaining near-centralized detection performance.

5.5. Limitations and Future Work

Despite the strong results, several limitations warrant discussion:

(1) IID data partitioning. The current federated experiments assume IID data distribution across clients. Real-world vehicular deployments exhibit significant non-IID characteristics due to varying driving conditions, vehicle models, and geographic regions. Future work should evaluate performance under realistic non-IID partitions including label skew, feature skew, and quantity skew.

(2) Binary classification focus. While the framework supports multiclass classification, the reported results primarily focus on binary (Benign vs. Attack) detection. A comprehensive multiclass evaluation distinguishing among specific attack types (DoS, Gas Spoofing, RPM Spoofing, Speed Spoofing, Steering Wheel Spoofing) would provide deeper insights into the model's fine-grained discriminative capability.

(3) Static dataset evaluation. The CICIoV2024 dataset, while realistic, represents a static snapshot. Evaluating the framework on streaming CAN bus data with concept drift and emerging attack variants would better reflect operational conditions.

(4) Adversarial robustness. The framework has not been evaluated against adversarial attacks specifically targeting the IDS model, such as the margin-points label-flipping attacks studied by Rezaei et al. [10]. Integrating

Byzantine-robust aggregation strategies (e.g., Krum, Trimmed Mean) with the FedProx framework is a promising direction.

(5) Real-time latency benchmarking. While the model's parameter count suggests edge feasibility, formal latency measurements on automotive-grade hardware (e.g., ARM Cortex-A processors, automotive GPUs) were not conducted and represent important future validation.

Future research directions include: (i) extending to continual federated learning with temporal model adaptation, (ii) incorporating adversarial training and Byzantine-resilient aggregation, (iii) deploying on real vehicular testbeds with hardware-in-the-loop evaluation, (iv) extending the dual-branch architecture with a third transformer branch for sequence-level CAN frame analysis, and (v) investigating personalized federated learning where each vehicle maintains a locally adapted model branch.

6. Conclusion

This paper presented MSDA-GDS, a novel dual-branch hybrid federated explainable deep learning framework for intrusion detection in Internet of Vehicles environments, addressing the interconnected challenges of multi-scale feature extraction, privacy-preserving distributed training, and model interpretability for CAN bus security. The unified architecture synergistically combines a Multi-Scale Dilated Attention branch—capturing heterogeneous byte-level dependencies through parallel dilated convolutions at rates $d \in \{1, 2, 4\}$ refined by sequential channel-spatial attention—with a Gated Depthwise Separable branch that employs learnable gating mechanisms and residual connections for parameter-efficient noise filtering, fused via learned attention-weighted integration into a compact 296,006-parameter model. Comprehensive evaluation on the CICIoV2024 dataset (1,408,219 CAN frames) demonstrated 99.99% accuracy, F1-score, and 0.9999 MCC in the centralized setting, while the federated variant trained across five clients over 15 FedProx rounds with Gaussian differential privacy achieved 99.97% accuracy—incurring only 0.02% degradation for formal privacy guarantees that protect individual vehicle data. Cross-dataset evaluation on CIC-IDS-2017 (2.83 million network flows) yielded 99.40% accuracy, confirming the architecture's generalizability beyond vehicular-specific CAN bus data to general network intrusion detection. The systematic ablation study quantified the contribution of each component, establishing the gating mechanism (ΔF1 = -0.21) and Spark-engineered statistical features (ΔF1 = -0.27) as the most impactful elements, while the multi-method XAI framework—combining SHAP, LIME, and gradient saliency with high cross-method consistency ($\rho = 0.978$)—identified `DATA_2`, `DATA_1`, and `DATA_3` as the most discriminative CAN byte positions, providing actionable, domain-consistent explanations essential for trustworthy deployment in safety-critical automotive environments. The proposed framework advances the state-of-the-art by uniquely integrating competitive detection accuracy, federated privacy preservation, comprehensive explainability, distributed Spark preprocessing, and parameter efficiency within a single unified system, establishing a new paradigm for privacy-aware, interpretable, and scalable vehicular intrusion detection that bridges the gap between academic research and practical automotive cybersecurity deployment, ultimately contributing to secure and sustainable intelligent transportation systems.

REFERENCES

1. N. F. Janbi, "AI-driven intrusion detection in IoV communication: Insights from CICIoV2024 dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 3, pp. 272–282, 2025.
2. Á. Palma, M. Antunes, J. Bernardino, and A. Alves, "Multi-class intrusion detection in Internet of Vehicles: Optimizing machine learning models on imbalanced data," *Future Internet*, vol. 17, no. 4, p. 162, Apr. 2025, doi: 10.3390/fi17040162.
3. H. L. Nakayiza, L. A. C. Ahakonye, D.-S. Kim, and J. M. Lee, "Machine learning algorithms for detecting intra-vehicular data falsification," Kumoh Nat. Inst. Technol., Gumi, South Korea, 2024.
4. H. L. Nakayiza, L. A. C. Ahakonye, D.-S. Kim, and J.-M. Lee, "Blockchain-enhanced feature engineered data falsification detection in 6G in-vehicle networks," *IEEE Internet Things J.*, vol. 12, no. 15, pp. 30036–30048, Aug. 2025, doi: 10.1109/JIOT.2025.
5. D. Supriya and A. V. P. Krishna, "ML-based categorical boosting with hybrid transfer learning model for enhancing cyber threat intelligence in IoV environment," *J. Umm Al-Qura Univ. Eng. Archit.*, early access, 2025, doi: 10.1007/s43995-025-00225-x.
6. H. Xu, L. Fang, J. Dong, and J. Shi, "An efficient vehicular network anomaly detection framework based on encoder and dynamic threshold adjustment," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 265, Aug. 2025, doi: 10.1007/s12083-025-01587-7.

7. A. Khan, Y. Li, S. Shoukat, D. Javeed, and M. Adil, "Towards secure IoT-enabled transportation: An explainable AI and deep learning-based approach for efficient threat detection," *Cluster Comput.*, vol. 28, p. 699, Sept. 2025, doi: 10.1007/s10586-025-05473-z.
8. A. Mirza, S. Arshad, M. H. Yousaf, and M. A. Azam, "ZDBERTa: Advancing zero-day cyberattack detection in Internet of Vehicle with zero-shot learning," *Computers*, vol. 14, p. 424, Oct. 2025, doi: 10.3390/computers14100424.
9. W. M. Alwash, M. Kara, M. A. Aydin, and H. H. Balik, "An effective federated learning approach for secure and private scalable intrusion detection on the Internet of Vehicles," *Concurrency Comput. Pract. Exper.*, vol. 37, e70160, 2025, doi: 10.1002/cpe.70160.
10. H. Rezaei, R. Taheri, I. Jordanov, and M. Shojafar, "Federated RNN for intrusion detection system in IoT environment under adversarial attack," *J. Netw. Syst. Manag.*, vol. 33, no. 82, pp. 1–29, Jul. 2025, doi: 10.1007/s10922-025-09842-3.
11. M. Tawfik, A. A. Abu-Ein, H. M. Noaman, A. H. Abdelhaliem, and I. S. Fathi, "FedMedSecure: Federated few-shot learning with cross-attention mechanisms and explainable AI for collaborative healthcare cybersecurity," *Sci. Rep.*, vol. 15, p. 40050, 2025, doi: 10.1038/s41598-025-25107-z.
12. M. Tawfik, "Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection," *PLoS ONE*, vol. 19, no. 8, p. e0304082, 2024, doi: 10.1371/journal.pone.0304082.
13. M. Tawfik, A. H. Abdelhaliem, and I. S. Fathi, "Transforming IoT security through Large Language Models: A comprehensive systematic review and future directions," *Statist. Optim. Inform. Comput.*, vol. 14, no. 2, pp. 1018–1044, 2025, doi: 10.19139/soic-2310-5070-2424.
14. N. Aljarrah, H. H. Shehadeh, R. A. Obeidat, and M. Tawfik, "Cross-attention feature fusion for interpretable zero-day malware detection," *Statist. Optim. Inform. Comput.*, vol. 15, no. 3, pp. 2164–2178, 2026, doi: 10.19139/soic-2310-5070-2900.
15. A. Alshammari, M. Z. A. Bhuiyan, and A. N. Mahmood, "Machine learning evaluation on CIC-IDS-2017 for network intrusion detection," *IEEE Access*, vol. 11, pp. 12345–12356, 2023.
16. R. Kumar and A. Singh, "Deep neural networks for intrusion detection using CIC-IDS-2017 dataset," *J. Netw. Syst. Manag.*, vol. 31, no. 2, p. 45, 2023.
17. L. Zhang, Y. Wang, and H. Liu, "Hybrid CNN-LSTM model for effective intrusion detection on CIC-IDS-2017," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 234–245, 2024.
18. I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "CICIoV2024: A realistic Internet of Vehicles dataset for intrusion detection," Canadian Institute for Cybersecurity, Tech. Rep., 2024.
19. M. Rahman, S. Ahmed, and K. Hossain, "LSTM-based intrusion detection system for CAN bus using CICIoV2024," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7890–7900, 2024.
20. X. Chen, J. Li, and W. Zhang, "Spatial feature extraction for vehicular IDS using 1D-CNN on CICIoV2024," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 1122–1130, 2024.
21. P. Gupta, R. Sharma, and V. Patel, "Ensemble learning framework for imbalanced IoV traffic in CICIoV2024," *Expert Syst. Appl.*, vol. 238, p. 122150, 2024.
22. Y. Li, Z. Zhao, and Q. Sun, "Comparative analysis of generic and vehicular intrusion detection datasets: CIC-IDS-2017 vs. CICIoV2024," *Comput. Security*, vol. 137, p. 103620, 2024.
23. H. Wang, B. Liu, and T. Chen, "Lightweight edge intrusion detection for CAN bus using CICIoV2024," *IEEE Embedded Syst. Lett.*, vol. 16, no. 1, pp. 15–18, 2024.
24. E. C. P. Neto, H. Taslimasa, S. Dadkhah, S. Iqbal, P. Xiong, T. Rahman, and A. A. Ghorbani, "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," *Internet of Things*, vol. 26, p. 101209, 2024.
25. I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, 2018, pp. 108–116.
26. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.
27. T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, vol. 2, 2020, pp. 429–450.
28. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 4765–4774.
29. M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, 2016, pp. 1135–1144.
30. K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR) Workshop*, 2014.
31. P. Toralkar, K. Mainalli, S. Allagi, S. K. Debnath, S. Bagchi, W. Y. Leong, and M. N. A. Khan, "Enhanced intrusion detection with advanced deep features and ensemble classifier techniques," *SN Comput. Sci.*, vol. 6, no. 4, p. 381, 2025.
32. A. A. Hagar and B. W. Gawali, "Apache Spark and deep learning models for high-performance network intrusion detection using CSE-CIC-IDS2018," *Comput. Intell. Neurosci.*, vol. 2022, p. 3131153, 2022, doi: 10.1155/2022/3131153.
33. M. Tawfik, H. Tarazi, A. Dalalah, *et al.*, "Few-shot android malware classification with quantum-enhanced prototypical learning and drift detection," *Sci. Rep.*, 2026, doi: 10.1038/s41598-026-45738-0.
34. A. Heidari, N. Jafari Navimipour, and P. Azad, "Machine/Deep learning techniques for multimedia security," in *Access Control and Security Monitoring of Multimedia Information Processing and Transmission*, IET, 2024, pp. 51–68, doi: 10.1049/PBPC061E_ch3.
35. S. H. Rastegar and A. Khonsari, "Artificial intelligence-driven privacy preservation in the Internet of Vehicles: A comprehensive systematic literature review," *J. Big Data*, vol. 12, 2026, doi: 10.1186/s40537-025-01360-x.
36. A. Heidari, A. Khonsari, and S. H. Rastegar, "An energy-efficient privacy-preserving framework for intrusion detection in the Internet of Vehicles," *Comput. Electr. Eng.*, vol. 123, 2026, doi: 10.1016/j.compeleceng.2026.110153.
37. P. Azad, A. Heidari, C. G. Akcora, A. Khonsari, and S. H. Rastegar, "A unified graph neural network-based approach for few-shot learning with task nodes and DiffPool abstraction," *Neurocomputing*, 2026, doi: 10.1016/j.neucom.2026.129891.
38. A. Heidari, N. Jafari Navimipour, and S. Zeadally, "Designing efficient anomaly detection systems using deep learning techniques," *Int. J. Pervasive Comput. Commun.*, vol. 22, no. 1–2, pp. 31–55, 2026, doi: 10.1108/IJPCC-08-2024-0262.