Privacy-Preserving Vehicle Intrusion Detection System Using Federated Learning and Homomorphic Encryption

Chloe Soriano de Leon* and Cedric Angelo Festin

Department of Computer Science, College of Engineering,
University of the Philippines Diliman, Quezon City, Philippines
*Corresponding author: csdeleon1@up.edu.ph

Abstract. As vehicles become more connected and autonomous, intrusion detection systems must adapt to emerging threats while preserving user privacy. This paper presents a privacy-preserving vehicle IDS that integrates federated learning (FL) and homomorphic encryption (HE) to detect denial-of-service, fuzzy, and impersonation attacks on Controller Area Network (CAN) traffic. Experiments were conducted on a publicly available CAN Intrusion Dataset using three models: Decision Tree (DT), Support Vector Classifier (SVC), and K-Nearest Neighbors (KNN). FL enables decentralized model training without exposing raw data, while CKKS-based HE secures encrypted aggregation. Results show that the FL+HE system maintains high detection accuracy with reasonable runtime overhead, making it suitable for offline or near-real-time diagnostic applications in privacy-sensitive vehicular environments.

Keywords: vehicle intrusion detection system, federated learning, homomorphic encryption, cybersecurity, data privacy, connected vehicles

1 Introduction

The evolution of connected and autonomous vehicles is transforming modern cars into intelligent, networked systems capable of decision-making and communication. Technologies such as advanced driver-assistance systems (ADAS) and vehicle-to-everything (V2X) enable data exchange between vehicles, infrastructure, and the cloud, improving safety, traffic flow, and user experience [1].

However, this increased connectivity exposes vehicles to new cybersecurity threats, with potentially severe safety, financial, and reputational consequences. High-profile incidents, such as the 2015 Jeep Cherokee hack that allowed remote disabling of steering and braking, prompted the recall of 1.4 million vehicles and cost Fiat Chrysler millions of dollars in remediation [2].

The Controller Area Network (CAN) bus, which links critical electronic control units (ECUs) such as those managing braking and steering, lacks built-in encryption or authentication, making it a common attack surface [3]. Notable threats include denial-of-service (DoS) attacks, which flood the network with

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2025)*, Atlantis Highlights in Computer Sciences 24,
https://doi.org/10.2991/978-94-6239-638-8_31

high-priority messages to disrupt communication; fuzzy attacks, which inject random or malformed frames; and impersonation attacks, which spoof legitimate ECUs to gain unauthorized control [4].

To counter these threats, intrusion detection systems (IDS) are employed to monitor CAN traffic for anomalies. Yet, traditional IDS architectures often rely on centralized data processing, which limits scalability and raises privacy concerns as vehicle data grows more sensitive and distributed [5].

This study proposes a privacy-preserving IDS that combines federated learning (FL) and homomorphic encryption (HE) to detect CAN-based attacks. While FL and HE have been explored in privacy-sensitive domains such as IoT and healthcare [6], their integration in vehicular IDS remains limited. FL enables decentralized training across vehicles without exposing raw data, while HE ensures encrypted aggregation of model updates. Together, they offer end-to-end privacy protection.

Despite introducing computational overhead and latency, the FL+HE approach retains detection performance on par with centralized systems. Our experiments show that it effectively identifies DoS, fuzzy, and impersonation attacks while supporting privacy-preserving, scalable deployment. The system is particularly suited for offline or near-real-time applications, such as post-drive diagnostics, where data confidentiality outweighs the need for immediate response.

2 Related Work

The rise in vehicular connectivity has elevated cybersecurity risks, driving the need for IDS that monitor in-vehicle networks for malicious activity. Several IDS approaches have emerged, each with its own trade-offs.

2.1 Traditional IDS Approaches

CAN bus systems, while critical to vehicle control, lack built-in security, making them vulnerable to attacks such as spoofing and message injection [3,7]. Traditional IDS implementations often use centralized architectures, where raw data is transmitted to remote servers for analysis. Detection methods are typically either signature-based, which struggle with unknown threats, or anomaly-based, which require large volumes of labeled data and are prone to false positives.

Both approaches suffer from:

1. **Privacy risks**: Centralized data collection may expose driver behavior, location, and usage patterns if intercepted [8,9].
2. **Scalability constraints**: Centralized systems become bottlenecked as vehicle fleets grow [10].
3. **Communication overhead**: Continuous transmission burdens network resources, especially in low-connectivity areas.

To address these limitations, recent work has shifted toward decentralized, anomaly-based approaches using machine learning models such as Support Vector Machines, Decision Trees, and k-Nearest Neighbors, which have shown promise in identifying novel attacks [11,12].

2.2 Federated Learning in IDS

Federated Learning (FL) enables decentralized training by keeping data on local clients. Each client trains a model on its own data and shares only updates with a central server, where a global model is formed via aggregation. This setup preserves privacy and supports heterogeneous environments. Figure 1 illustrates the FL process in vehicle IDS.

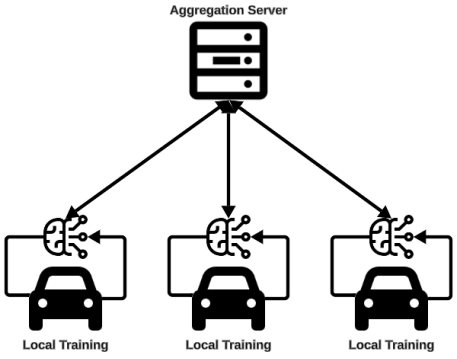


Fig. 1: FL framework for vehicle IDS. Local training occurs on individual vehicles; model updates are aggregated at a central server without exposing raw data.

Prior work has applied FL to IDS with promising results. Xing et al. [13] used FL with memory-augmented autoencoders, while Sebastian et al. [14] applied SMOTE and outlier detection to improve class balance. However, FL remains susceptible to inference attacks, where shared gradients can leak sensitive data [15].

2.3 Homomorphic Encryption in IDS

Homomorphic Encryption (HE) addresses this gap by enabling computation on encrypted data. In FL, clients encrypt model updates, which are aggregated by the server without decryption. Only the clients can decrypt the final result.

HE schemes are built on algebraic structures such as polynomial rings and modular arithmetic. For example, if two plaintext messages m_1 and m_2 are encrypted to ciphertexts c_1 and c_2, then:

$$\mathrm{Enc}(m_1) + \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 + m_2),\ \mathrm{Enc}(m_1) \times \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 \times m_2)$$

Noise accumulates with operations, and techniques such as bootstrapping or relinearization are used to manage this [16].

Depending on the operations supported, HE is categorized as Partial (PHE), Somewhat (SHE), or Fully Homomorphic Encryption (FHE). Table 1 compares these categories.

Table 1: Comparison of Homomorphic Encryption Types

Type	**Addition**	**Multiplication**	**Operation Depth**	**Best Use Case**
PHE	Yes or No	Yes or No	Unlimited (one operation only)	Secure voting, simple sums
SHE	Yes	Yes	Limited	Shallow models, single-step processing
FHE	Yes	Yes	Unlimited	Deep learning, full encrypted computation

Among FHE schemes, BFV and BGV support exact integer arithmetic, while CKKS allows approximate computation on real numbers, ideal for machine learning tasks [17,18,19]. Table 2 summarizes their distinctions.

Table 2: Comparison of Common FHE Schemes

Scheme	**Data Type**	**Operation Type**	**Use Case**
BFV	Integer	Exact	Encrypted integer computations
BGV	Integer (batching)	Exact	Batched encrypted computations
CKKS	Real/Complex (approximate)	Approximate	Machine learning, encrypted model training

This work uses CKKS due to its support for efficient floating-point operations. Supported libraries include SEAL, TenSEAL, and HElib [20,21,22]. While HE is well-suited to secure computation in cloud and healthcare applications [23], it introduces performance overhead in IDS deployments [8,24,25].

2.4 Combining Federated Learning and Homomorphic Encryption

Combining FL with HE yields end-to-end privacy: raw data stays local via FL, while HE secures model updates during aggregation [9]. Clients train locally, encrypt updates with CKKS, and send them to a server that aggregates them

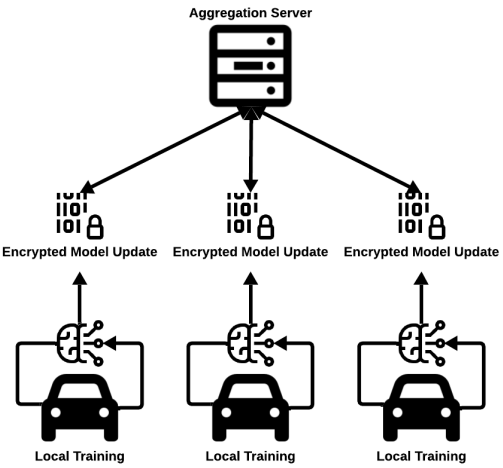


Fig. 2: Combined Federated Learning and Homomorphic Encryption framework. Clients train local models, encrypt updates, and the server aggregates encrypted updates without accessing plaintext data.

without decryption. The encrypted global model is returned and decrypted only by clients. Figure 2 illustrates this process.

Recent work supports this integration. Jin et al. [10] proposed FedML-HE for selective encryption. Arazzi et al. [26] used FL+HE with blockchain for IoT security. ChandraUmakantham et al. [27] showed that encrypted FL enhances IDS privacy with minimal accuracy loss.

Despite growing interest, few works apply FL+HE to vehicular IDS. This study addresses that gap by presenting a privacy-preserving IDS using FL for decentralized training and CKKS-based HE for secure aggregation, tested on real-world CAN data.

3 System Overview

This system simulates a single round of federated aggregation to evaluate the impact of HE in a collaborative vehicle IDS. Each client represents a data partition simulating a vehicle, locally training a model on its CAN subset and generating predictions on a shared test set.

The CAN bus facilitates ECU communication by transmitting timestamped messages containing IDs and data payloads. As shown in Figure 3, the IDS ECU monitors for anomalies, such as repeated low-priority IDs or malformed payloads, that indicate DoS or fuzzy attacks. Due to the lack of encryption and authentication in CAN, attackers can easily inject or spoof messages, making IDS crucial for vehicular cybersecurity.

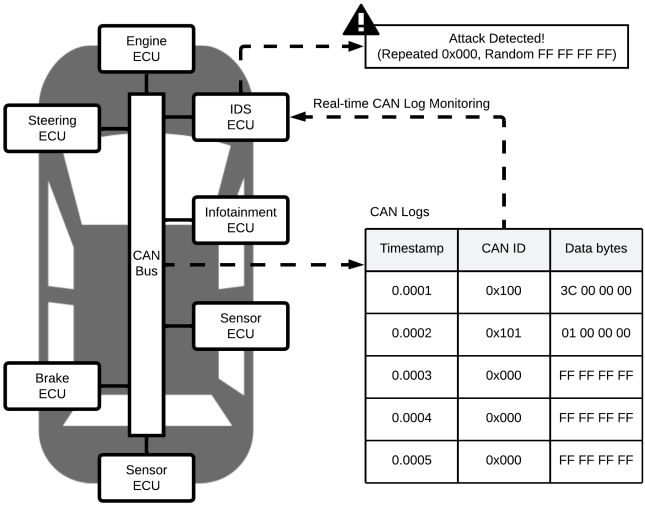


Fig. 3: Vehicle CAN bus architecture. ECUs exchange control messages; the IDS ECU logs and analyzes traffic to detect intrusions.

Figure 4 illustrates the system architecture, which integrates FL, HE, and server-side aggregation. Clients perform local training and inference, encrypt prediction vectors, and transmit them for secure aggregation. Only the final result is decrypted.

The system assumes vehicles have embedded systems capable of running lightweight machine learning and encryption, along with secure links (e.g., DSRC, C-V2X) for periodic model updates. While HE adds processing overhead, using CKKS and limiting encryption to prediction aggregation keeps runtime practical. In real deployments, compute-intensive tasks can be offloaded to edge gateways (e.g., roadside units) to enhance scalability.

The system consists of the following components:

- **Clients**: Simulated as separate data partitions, each client (vehicle) trains local models on CAN data and produces predictions.
- **Central Server**: Aggregates predictions and decrypts only final results in the FL+HE setting.
- **Federated Learning Module**: Coordinates local training, data splitting, and collaborative prediction through majority voting or encrypted aggregation.
- **Homomorphic Encryption Layer**: Applies CKKS encryption to client predictions, enabling privacy-preserving aggregation.

4 Dataset Description

This study uses the publicly available CAN Intrusion Dataset from the Hacking and Countermeasure Research Lab, Korea University [28]. The dataset contains

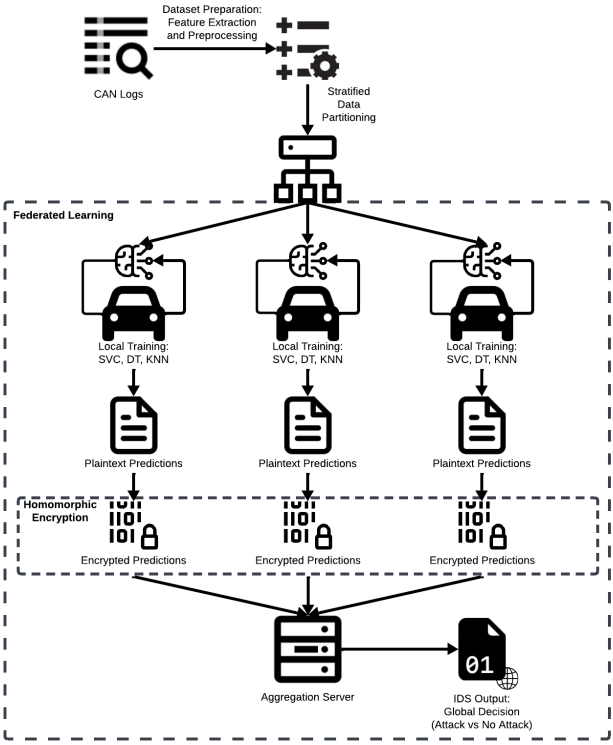


Fig. 4: FL+HE architecture. Clients perform local inference; encrypted predictions are aggregated by the server.

real-world CAN traffic logs collected from a production car under normal and attack scenarios, with one benign and three attack subsets: DoS, Fuzzy, and Impersonation. Each subset contains timestamped CAN frames with a message ID, data length, and up to 8 payload bytes. The original raw dataset includes:

- **Attack-Free Dataset:** 2,369,398 lines
- **DoS Attack Dataset:** 656,579 lines
- **Fuzzy Attack Dataset:** 591,990 lines
- **Impersonation Attack Dataset:** 995,472 lines

After parsing and preprocessing, we obtained **4,613,439 labeled samples** with the following class distribution:

- **Normal (Label 0)**: 2,369,398 samples
- **Attack (Label 1)**: 2,244,041 samples

Feature extraction was guided by the work of Bari et al. [29], focusing on behavioral and timing anomalies commonly exploited in CAN-based attacks. The selected features include:

- **Timestamp**: Detects message floods common in DoS and Fuzzy attacks.
- **Last Remote Timestamp**: Measures time since last remote request.
- **Data Size (DLC)**: Identifies abnormal payload lengths in Fuzzy attacks.
- **Remote Offset, Remote Response Time, Remote Interval**: Capture irregular timing patterns associated with impersonation or flooding.

These features correspond to known attack behaviors: DoS floods the bus with repeated IDs, Fuzzy injects malformed data, and Impersonation mimics legitimate IDs while disrupting expected response timing.

To ensure balanced evaluation, the dataset was sorted by timestamp, normalized, and stratified for training and federated client partitioning.

5 Model Training and Federated Learning Setup

This study evaluates three commonly used models for CAN intrusion detection: Support Vector Classifier (SVC), Decision Tree (DT), and K-Nearest Neighbors (KNN). These models were chosen for their effectiveness in IDS tasks [30,31], interpretability [32,33], and suitability for resource-constrained devices [34,35]. Future work will explore other lightweight machine-learning models that remain compatible with the privacy-preserving design.

5.1 Model Configurations

- **SVC**: Linear kernel for high-dimensional separation. We set `max_iter=10000`, `dual=False` to ensure convergence on our wide dataset [36,37].
- **DT**: Provides interpretable rules. To reduce overfitting, we limit tree depth to 15 and require 50 samples per split [38,39].
- **KNN**: We used `n_neighbors=5` with `algorithm='kd_tree'` for efficient search in large datasets [40,41].

5.2 Federated Learning Setup

For FL and FL+HE experiments, training data was split across simulated clients using stratified sampling to preserve the class distribution (∼51% normal, ∼49% attack). Each client trains locally and predicts on a shared test set. The server aggregates predictions as follows:

- **FL**: Majority voting on plaintext predictions.
- **FL+HE**: Clients encrypt predictions using CKKS; the server aggregates ciphertexts and decrypts only the final result.

5.3 Training Configuration

We simulate a single communication round to isolate the impact of encrypted inference. Fixed hyperparameters were selected based on prior literature and preliminary tuning [42,43], balancing performance and feasibility under limited server resources.

Stratified partitioning and timestamp-ordered sampling were applied to maintain the natural class ratio and to prevent data leakage between training and test partitions. This approach ensures that similar CAN frames are not duplicated across splits, reducing the risk of overfitting while preserving temporal behavior of the traffic.

5.4 Homomorphic Encryption Parameters

Encrypted aggregation was implemented using the CKKS scheme with the following parameters:

- **Polynomial Modulus Degree**: `4096` balances security and speed (128-bit level) [44]
- **Coefficient Modulus Bit Sizes**: `[30, 20, 20, 30]` manages noise growth [45].
- **Global Scale**: 2^{20} provides good floating-point precision without excessive noise [46]

These settings align with best practices in SEAL and other FHE libraries.

6 Evaluation Setup

6.1 Train-Test Split

We used a stratified 75/25 split to separate the dataset into training and test sets, maintaining the original class ratio ($\sim$51% normal, $\sim$49% attack). The resulting sizes were:

- Training set: $\sim$3.46 million samples
- Test set: $\sim$1.15 million samples

Only the test set was used for evaluation to ensure fair generalization.

6.2 Model Evaluation Metrics

Models were evaluated using:

- **Accuracy**: Percentage of correctly predicted samples.
- **Inference Time**: Time taken to generate predictions. For encrypted setups, this includes encryption, aggregation, and decryption.

6.3 Evaluation Scenarios

To assess the impact of FL and HE, we tested four configurations:

- **Baseline**: Centralized training and inference with no encryption or federation
- **FL**: Federated model training with no encryption
- **HE**: Centralized training with encrypted inference
- **FL+HE**: Federated training with encrypted inference and aggregation

6.4 Sample Sizes and Client Counts

To assess performance at scale, we trained models using 50,000 and 100,000 samples and varied the number of simulated clients: 1 (centralized), 3, 5, 10, and 20. Stratified partitioning was used in all cases to preserve class balance per client.

6.5 Runtime Logging and Output

For each model–setup combination, we recorded training time, inference time, total runtime, and accuracy, totaling 120 configurations. Results were saved in CSV format and visualized using line plots for comparison.

6.6 Tools and Libraries

Experiments were implemented in Python using:

- `scikit-learn` for model training
- `TenSEAL` for CKKS-based encryption
- `pandas`, `time`, and `logging` for tracking
- `matplotlib` for visualization

6.7 Scalability Limitations

A trial run using 200,000 samples and 40 clients was attempted on the UP Diliman GPU server but was terminated due to time constraints:

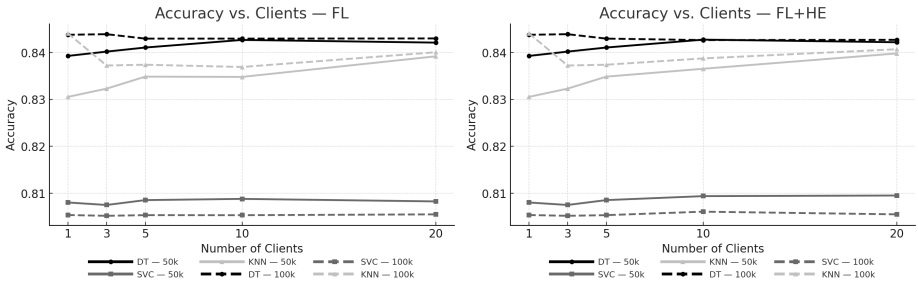
```
slurmstepd: error: *** JOB 79690 ON a100gpu1 CANCELLED AT
2025-04-14T19:19:46 DUE TO TIME LIMIT ***
```

This shows the system scales moderately well, but larger deployments may require longer runtimes, better encryption pipelines, or distributed strategies.

7 Results and Discussion

7.1 Overview of Experimental Results

We evaluated SVC, DT, and KNN models under four configurations (Baseline, FL, HE, FL+HE), two training sizes (50k, 100k), and five client counts (1, 3, 5, 10, 20), totaling 120 runs. We measured test accuracy and inference time.



(a) FL configuration (b) FL+HE configuration

Fig. 5: Accuracy vs. number of clients for FL and FL+HE configurations (50k and 100k samples).

7.2 Accuracy and Runtime Trends

Figures 5(a)–(b) show the accuracy of the federated configurations (FL and FL+HE) using 50k and 100k samples. Figures 6(a)–(d) illustrate inference time under each setup.

Accuracy remained stable across all configurations. DT achieved the highest (up to 84.4%), benefiting from rule-based detection aligned with CAN traffic behavior [38,30]. KNN also performed well with larger datasets, although it was more sensitive to client splits. SVC had the most consistent performance but slightly lower accuracy (∼80.5%) due to its linear nature [36,47].

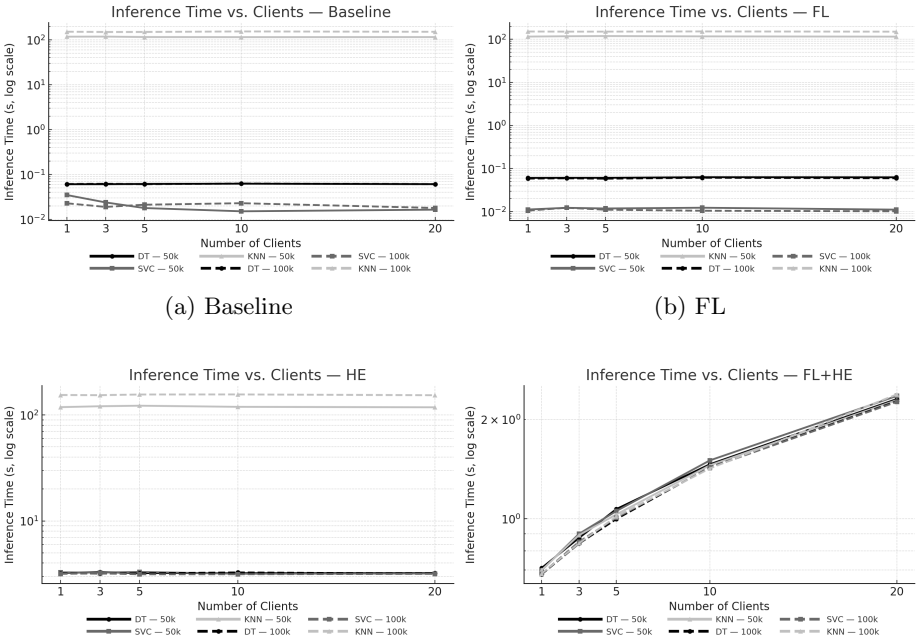
Client-count comparisons apply only to the federated configurations (FL and FL+HE), since FL distributes model training across multiple clients. In contrast, Baseline and HE are centralized setups that use a single trained model for inference, so accuracy remains constant regardless of the number of clients.

More data improved accuracy by up to 1.3%, confirming model scalability without additional privacy costs. KNN saw the largest gain; SVC remained unchanged, showing it plateaued early due to its linear nature.

Inference time varied widely. Baseline and FL executed predictions in under 0.1s. HE introduced the largest overhead (≈3s) due to encrypted operations, whereas FL+HE significantly reduced runtime by encrypting only prediction vectors during aggregation.

Unlike the other configurations, FL+HE shows a noticeable upward trend in inference time as client count increases. This behavior reflects the additional encryption and aggregation overhead incurred when combining encrypted prediction vectors from multiple clients. The increase, however, remains sub-linear and well below the full HE runtime, demonstrating that FL+HE achieves strong privacy with practical efficiency.

The runtime gap between Baseline and FL+HE remains small enough to support practical offline or near-real-time diagnostics.



(a) Baseline (b) FL

(c) HE (d) FL+HE

Fig. 6: Inference time vs. number of clients (log scale) for all configurations. HE introduces the highest overhead; FL+HE reduces runtime while preserving accuracy.

Notably, KNN's inference time dropped from ∼130s (Baseline) to ∼1.3s (FL+HE) due to encrypted prediction averaging [40,48].

7.3 Limitations

- **Simulated environment**: Real-world vehicular networks involve unpredictable conditions and external interactions not captured here.
- **Focus on digital threats only**: Physical security threats, such as hardware tampering, are out of scope.
- **Not suitable for real-time detection**: HE overhead and FL aggregation latency limit use to offline or periodic diagnostics.

7.4 Summary of Findings

FL+HE supports privacy-preserving detection with moderate runtime and high accuracy, suitable for offline diagnostics in connected vehicles. DT is the best fit overall: accurate, fast, and interpretable.

Table 3 summarizes the privacy, performance, and runtime trade-offs across the evaluated configurations.

Table 3: Summary of Configuration Trade-Offs

Configuration	Privacy	Accuracy	Inference Time	Notes
Baseline	None	80-84%	DT, SVC: ∼0.01-0.06 sec KNN: ∼118-154 sec	Centralized No privacy protection
FL	Moderate	80-84%	DT, SVC: ∼0.01-0.06 sec KNN: ∼115-153 sec	Decentralized Raw updates exposed
HE	Strong	80-84%	DT, SVC: ∼3 sec KNN: ∼118-156 sec	Encrypted inference Centralized training
FL+HE	Strongest	80-84%	All models: ∼0.7-2.4 sec	Fully decentralized Encrypted aggregation

8 Conclusion and Future Work

8.1 Conclusion

This study presented a privacy-preserving vehicle IDS that combines FL and HE to detect CAN-based cyberattacks. Using real-world data and simulated clients, the system achieved high detection accuracy while safeguarding sensitive information during both training and inference.

DT achieved the best accuracy, while SVC remained the most consistent. Despite added runtime from CKKS-based encryption, FL+HE preserved accuracy close to baseline models and enabled secure collaborative detection. These results demonstrate that privacy and performance can coexist in offline or near-real-time intrusion detection systems for connected vehicles.

8.2 Future Work

There are several directions to build on this work:

- **Multi-round FL:** Evaluate model convergence, drift, and communication costs in iterative FL.
- **Multi-class Detection:** Extend beyond binary labels to identify specific attack types or anomaly severity.
- **Hardware Testing:** Deploy the system on embedded devices or in-vehicle simulators for practical evaluation.
- **Optimizing HE:** Explore ciphertext packing, hardware acceleration, or parameter tuning to reduce encryption overhead [49,50,51].
- **Toward Real-Time IDS:** Investigate lightweight HE variants (e.g., TFHE) or edge-assisted inference to support low-latency detection [52,53].

Acknowledgement

This research was supported in part by the Cesar A. Buenaventura Distinguished Professorial Chair. The authors also acknowledge the High Performance Computing (HPC) facility of University of the Philippines Diliman for the computational resources used in this study.

References

1. Md Masud Rana and Kamal Hossain, "Connected and autonomous vehicles and infrastructures: A literature review," *International Journal of Pavement Research and Technology*, vol. 16, no. 2, pp. 264–284, 2023, Springer.
2. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. In: Black Hat USA (2015). https://ioactive.com/pdfs/IOActive_Remote_Car_Hacking.pdf
3. Damilola Oladimeji, Amar Rasheed, Cihan Varol, Mohamed Baza, Hani Alshahrani, and Abdullah Baz, "CANAttack: Assessing Vulnerabilities within Controller Area Network," *Sensors*, vol. 23, no. 19, p. 8223, 2023, MDPI.
4. Mahender Reddy Bobbala and R. Kavitha, "Analyzing The Can Protocol: Vulnerabilities, Protective Measures and Improvements," *International Journal of Interpreting Enigma Engineers (IJIEE)*, vol. 1, no. 1, pp. 10–15, 2024.
5. Arash Heidari and Mohammad Ali Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," *Cluster Computing*, vol. 26, no. 6, pp. 3753–3780, 2023, Springer.
6. Yuhang Liu, Rongxing Zhang, Tian Yu, and Yulei Liu, "Privacy-Preserving Collaborative Learning with Homomorphic Encryption: Applications in IoT and Healthcare," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14567–14579, 2022, doi: 10.1109/JIOT.2022.3144996.
7. Junaid Khan, Dae-Woon Lim, and Young-Sik Kim, "Intrusion detection system CAN-BUS in-vehicle networks based on the statistical characteristics of attacks," *Sensors*, vol. 23, no. 7, p. 3554, 2023, MDPI.
8. Anca Hangan, Dragos Lazea, and Tudor Cioara, "Privacy Preserving Anomaly Detection on Homomorphic Encrypted Data from IoT Sensors," *arXiv preprint arXiv:2403.09322*, 2024.
9. Neveen Mohammad Hijazi, Moayad Aloqaily, Mohsen Guizani, Bassem Ouni, and Fakhri Karray, "Secure federated learning with fully homomorphic encryption for IoT communications," *IEEE Internet of Things Journal*, 2023, IEEE.
10. Weizhao Jin, Yuhang Yao, Shanshan Han, Carlee Joe-Wong, Srivatsan Ravi, Salman Avestimehr, and Chaoyang He, "FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System," *arXiv preprint arXiv:2303.10837*, 2023.
11. Asim Raza Javed, Muhammad Othman Beg, Nauman Aslam, and Thar Baker, "Autoencoder-based Anomaly Detection for In-Vehicle Networks: A Deep Learning Approach," *Computers & Security*, vol. 117, p. 102693, 2022, Elsevier. doi:10.1016/j.cose.2022.102693
12. Zheng Zhang, Rui Huang, Yuming Lu, and Zhiyong Liu, "A Lightweight Anomaly Detection Method for In-Vehicle Networks Based on Message Frequency," *IEEE Access*, vol. 9, pp. 122063–122072, 2021, IEEE. doi:10.1109/ACCESS.2021.3109301

13. Lin Xing, Kun Wang, Hui Wu, Haibo Ma, and Xiaolong Zhang, "FL-MAAE: An Intrusion Detection Method for the Internet of Vehicles Based on Federated Learning and Memory-Augmented Autoencoder," *Electronics*, vol. 12, no. 10, p. 2284, 2023, MDPI. doi:10.3390/electronics12102284
14. Abhishek Sebastian *et al.*, "Enhancing Intrusion Detection In Internet Of Vehicles Through Federated Learning," *arXiv preprint arXiv:2311.13800*, 2023.
15. Ligeng Zhu, Zhijian Liu, and Song Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 14774–14784, 2020.
16. Ivone Amorim and Ivan Costa, "Leveraging Searchable Encryption through Homomorphic Encryption: A Comprehensive Analysis," *Mathematics*, vol. 11, no. 13, p. 2948, 2023, MDPI.
17. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference - ITCS 12*, 2012. doi:10.1145/2090236.2090262
18. J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," *IACR Cryptology ePrint Archive*, 2012.
19. J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *Advances in Cryptology – ASIACRYPT 2017*, Lecture Notes in Computer Science, pp. 409–437, 2017. doi:10.1007/978-3-319-70694-8
20. Francisco-Jose Valera-Rodriguez, Pilar Manzanares-Lopez, and Maria-Dolores Cano, "Empirical Study of Fully Homomorphic Encryption Using Microsoft SEAL," *Applied Sciences*, vol. 14, no. 10, p. 4047, 2024, MDPI.
21. Yancho B Wiryen, Noumsi Woguia Auguste Vigny, Mvogo Ngono Joseph, and Fono Louis Aimé, "A Comparative Study of BFV and CKKs Schemes to Secure IoT Data Using TenSeal and Pyfhel Homomorphic Encryption Libraries," *International Journal of Smart Security Technologies (IJSST)*, vol. 10, no. 1, pp. 1–17, 2024, IGI Global.
22. Haoyun Zhu, Takuya Suzuki, and Hayato Yamana, "Performance Comparison of Homomorphic Encrypted Convolutional Neural Network Inference Among HElib, Microsoft SEAL and OpenFHE," in *2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–7, 2023, IEEE.
23. Thi Van Thao Doan, Mohamed-Lamine Messai, Gérald Gavin, and Jérôme Darmont, "A survey on implementations of homomorphic encryption schemes," *The Journal of Supercomputing*, vol. 79, no. 13, pp. 15098–15139, 2023, Springer.
24. Jing Wang, Zhuoqun Xia, Yaling Chen, Chang Hu, and Fei Yu, "Intrusion Detection Framework Based on Homomorphic Encryption in AMI Network," *Frontiers in Physics*, vol. 10, p. 1102892, 2022.
25. Khalil Ahamed, "Enhancing Privacy in Cloud Anomaly Detection with Lightweight Homomorphic Encryption," *International Journal of Computer Science & Information System*, vol. 8, no. 08, pp. 01–04, 2023.
26. Marco Arazzi, Serena Nicolazzo, and Antonino Nocera, "A fully privacy-preserving solution for anomaly detection in IoT using federated learning and homomorphic encryption," *Information Systems Frontiers*, pp. 1–24, 2023, Springer.
27. Om Kumar ChandraUmakantham, Sudhakaran Gajendran, and Suguna Marappan, "Enhancing Intrusion Detection through Federated Learning with Enhanced Ghost_BiNet and Homomorphic Encryption," *IEEE Access*, 2024, IEEE.
28. Hacking and Countermeasure Research Lab. CAN Intrusion Dataset. Dataset available at: https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset. 2020.

29. B. S. Bari, K. Yelamarthi, and S. Ghafoor, "Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study," *Electronics*, vol. 11, no. 3, p. 490, 2022.
30. Mohamed Amine Ferrag, Leandros Maglaras, Helge Janicke, Richard Smith, and Sokratis K. Katsikas, "Deep Learning for Cybersecurity Intrusion Detection: Approaches, Datasets, and Comparative Evaluation," *Computers & Security*, vol. 87, pp. 101773, 2020. doi:10.1016/j.cose.2019.101773
31. Ruhul Islam and Jemal H. Abawajy, "A Survey on Machine Learning Based Intrusion Detection Systems for IoT Applications," *Information Fusion*, vol. 55, pp. 85–101, 2020. doi:10.1016/j.inffus.2020.01.008
32. David Gunning and David W. Aha, "DARPA's Explainable Artificial Intelligence (XAI) Program," *AI Magazine*, vol. 40, no. 2, pp. 44–58, 2019. doi:10.1609/aimag.v40i2.2850
33. Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Andrés Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020. doi:10.1016/j.inffus.2019.12.012
34. Thanh Tu Nguyen, Dinh Thai Hoang, Dusit Nguyen, Eryk Dutkiewicz, and H. Vincent Poor, "Federated Learning for Attack Detection in Industrial IoT: Concepts, Challenges, and Opportunities," *IEEE Network*, vol. 35, no. 2, pp. 246–253, 2021. doi:10.1109/MNET.011.2000490
35. Yutao Lu, Xuyu Huang, Yannan Dai, Sabita Maharjan, and Yan Zhang, "Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020. doi:10.1109/TII.2019.2942190
36. Jeffrey Mill and Atsuyuki Inoue, "Support Vector Classifiers and Network Intrusion Detection," in *Proceedings of the 2003 IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 413–418, 2003, IEEE.
37. Fabian Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
38. S. Shilpashree and K. G. Suneel, "Decision Tree: A Machine Learning for Intrusion Detection," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 6S4, pp. 1126–1130, 2019.
39. Nolan Kuhl, "Decision Trees for the Classification of Malware Behavior," *Computers & Security*, vol. 94, pp. 101854, 2020.
40. Yung-Chi Liao and Vasant H. Vemuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002, Elsevier.
41. Jon L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
42. Yuxuan Wang *et al.*, "Resource-Constrained Machine Learning: Survey and Challenges," *IEEE Access*, vol. 8, pp. 101256–101270, 2020.
43. Shreya Chakraborty *et al.*, "A Survey of Federated Learning for Privacy-Preserving Training in Resource-Constrained Environments," *ACM Computing Surveys*, 2021.
44. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *Advances in Cryptology – ASIACRYPT 2017*, Lecture Notes in Computer Science, vol. 10624, pp. 409–437, 2017, Springer. doi:10.1007/978-3-319-70694-8_15
45. Microsoft Research, *Microsoft SEAL (release 4.1) Manual*, 2023. Available at: https://github.com/microsoft/SEAL

46. Florian Bourse, Marco Minelli, Matthias Minihold, and Pascal Paillier, "Practical Homomorphic Encryption over the Integers with Applications to Privacy-Preserving Machine Learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1321–1338, 2018, ACM. doi:10.1145/3243734.3243837
47. Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018, doi: 10.1109/TETCI.2017.2759239.
48. Garima Sharma, Inderveer Saini, and Ramesh Sayal, "An Efficient Hybrid Intrusion Detection System Based on CFS Feature Selection and Classification Techniques," *Egyptian Informatics Journal*, vol. 21, no. 1, pp. 23–31, 2020, doi: 10.1016/j.eij.2019.05.003.
49. Jung Hee Cheon, Minkyu Kim, Yongsoo Lee, and Minsoo Ryu, "Privacy-Preserving Genome-Wide Association Studies with Optimal Test Statistics," *BMC Medical Genomics*, vol. 12, suppl. 2, p. 44, 2019.
50. Youssef Elmehdwi, Charles Kamhoua, and Kevin Kwiat, "Efficient and Secure KNN Classification with Homomorphic Evaluation," in *Proc. IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 165–172, 2019.
51. Florian Bourse, Marco Minelli, Dung Phan, and Sameer Wagh, "Improving Performance of Fully Homomorphic Encryption Through Tensorization for Financial Prediction," *IEEE Transactions on Computers*, vol. 71, no. 7, pp. 1686–1699, 2022.
52. Ilaria Chillotti, Nicolas Gama, Mariana Georgieva, and Mehdi Izabachene, "TFHE: Fast Fully Homomorphic Encryption Over the Torus," *Journal of Cryptology*, vol. 33, pp. 34–91, 2020.
53. Jinsoo Kim, Yao Wu, Junbeom Choi, Yuriy Polyakov, and Kurt Rohloff, "Accelerating Privacy-Preserving Machine Learning with Hardware-Friendly Approximate Bootstrapping," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 360–377, 2022.