frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 02 November 2026
DOI 10.3389/frai.2026.1949474

Check for updates

OPEN ACCESS

EDITED BY
Jichao Bi,
Zhejiang Institute of Industry and Information Technology, China

REVIEWED BY
Nitin Goyal,
Central University of Haryana, India
Shashikant Patil,
Atlas SkillTech University, India

*CORRESPONDENCE
Asim Zeb
✉ asim.zeb@aust.edu.pk
Rostam Affendi Bin Hamzah
✉ rostamaffendi@utem.edu.my
Jamil Abedalrahim Jamil Alsayaydeh
✉ jamil@utem.edu.my

RECEIVED 27 July 2026
REVISED 21 September 2026
ACCEPTED 15 October 2026
PUBLISHED 02 November 2026

CITATION
Khan FM, Zeb A, Bacarra R, Alsayaydeh JAJ, Hamzah RAB and Mahmud M (2026) MoFed-Guard: a mobility-aware federated and explainable lightweight Transformer framework for misbehavior detection in VANETs. *Front. Artif. Intell.* 9:1949474. doi: 10.3389/frai.2026.1949474

COPYRIGHT
© 2026 Khan, Zeb, Bacarra, Alsayaydeh, Hamzah and Mahmud. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

MoFed-Guard: a mobility-aware federated and explainable lightweight Transformer framework for misbehavior detection in VANETs

Fida Muhammad Khan[1], Asim Zeb[2]*, Rex Bacarra[3], Jamil Abedalrahim Jamil Alsayaydeh[4]*, Rostam Affendi Bin Hamzah[4]* and Maqsood Mahmud[5]

[1]Department of Computer Science, Qurtuba University of Science and Information Technology, Peshawar, Pakistan, [2]Department of Computer Science, Abbottabad University of Science and Technology, Abbottabad, Pakistan, [3]Department of General Education and Foundation, Rabdan Academy, Abu Dhabi, United Arab Emirates, [4]Department of Engineering Technology, Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia, [5]School of Computing, Ulster University, Belfast, Northern Ireland, United Kingdom

Authenticated vehicles can disseminate false safety messages that pass cryptographic verification but remain behaviorally untruthful, making insider misbehavior a challenge in Vehicular *Ad Hoc* Networks (VANETs). This study proposes *MoFed-Guard*, a data-local federated framework combining *OBU-TFT*, a compact temporal Transformer, with reliability-aware aggregation and on-demand SHAP explanation. Fixed RSUs act as training clients and OBUs perform inference. MoFed-Guard weights client updates using vehicle-association stability, communication reliability, and leave-one-client-out update consistency; these indicators are evaluated empirically rather than treated as theoretically derived trust variables. On the trajectory-disjoint 20-class VeReMi Extension test partition, the framework achieved 95.20% accuracy, 94.75% macro-precision, 94.30% macro-recall, and 94.52% macro-F1. With architecture and training fixed, MoFed-Guard improved macro-F1 by 1.45 points over FedAvg and 0.61 points over FedAdam; the five-seed improvement was significant [$t(4) = 6.30$, $p = 0.003$, $d_z = 2.82$]. Five-seed convergence reached 98.0±0.2% validation macro-F1 at round 100, while held-out client-scale macro-F1 increased from 93.10% with three clients to 94.52% with ten. Explicit FL-layer delivery tests and a road-network-aware partition preserved the aggregation benefit, whereas coordinated-client corruption exposed the limitation of cosine-based consistency. The joint reliability score correlated positively with update utility ($\rho = 0.71$, $p < 0.001$). OBU-TFT contains 85,460 trainable parameters, occupies 0.34 MB, and achieved 5.10 ± 0.42 ms T4 inference latency. Parameter-matched comparison showed that OBU-TFT achieved higher macro-F1 than Linformer and Performer, although Linformer remained smaller and faster. These results support empirical reliability-conditioned federated weighting under simulation without implying universal convergence, Byzantine robustness, formal privacy, or field deployment readiness.

KEYWORDS
explainable artificial intelligence (SHAP), federated learning, lightweight Transformer, misbehavior detection, reliability-aware aggregation, Vehicular *Ad Hoc* Networks (VANETs)

1 Introduction

Cooperative Intelligent Transportation Systems (C-ITS) use Vehicular *Ad Hoc* Networks (VANETs) to exchange information among vehicles and roadside infrastructure for cooperative awareness, collision avoidance, platooning, and traffic management (Sharma and Jaekel, 2021; Bouchouia et al., 2023; Tariq, 2024). Vehicles periodically broadcast kinematic information, including position, speed, acceleration, and heading. However, VANETs' decentralized and rapidly changing communication environment enables authenticated vehicles to transmit false, manipulated, or contextually misleading safety messages (Gu et al., 2025; Alladi et al., 2023). Prior VANET studies also emphasize that rapidly changing topology, route instability, quality-of-service variation, and scalability constraints can strongly affect communication continuity and system behavior (Chauhan et al., 2016; Radaur, 2015; Rasheed et al., 2017). Cryptographic mechanisms can verify identity, integrity, and nonrepudiation, but they cannot determine whether a valid message truthfully represents the sender's behavior (Aloraini et al., 2024; Tyagi et al., 2024). Misbehavior detection is therefore required as a complementary security mechanism. Machine- and deep-learning methods can identify temporal, spatial, and behavioral anomalies that are difficult to represent with fixed plausibility rules (Bhukya et al., 2024; El-Gayar et al., 2024; Qu and Cai, 2024; Tariq and Ahanger, 2025; Rashid et al., 2023). However, many existing methods rely on centralized vehicular observations, substantial edge or cloud computation, or evaluation protocols that remain sensitive to class imbalance, traffic variation, topology changes, and intermittent connectivity. These limitations are particularly relevant to benchmarks such as VeReMi (Van Der Heijden et al., 2018), where data partitioning and class composition can strongly affect the reported performance. Federated learning (FL) supports collaborative training without centralizing raw observations (Singh et al., 2025); however, raw-data locality alone does not guarantee confidentiality because model updates may still disclose information unless additional privacy mechanisms are applied (Zhang et al., 2024).

FedAvg conventionally weights participating clients according to their local sample counts. Under non-independent and identically distributed (non-IID) data, unreliable links, and inconsistent local optimization, sample count may not accurately represent update utility. This limitation is important when fixed RSUs serve as federated clients but observe different, time-varying vehicle populations. In this setting, mobility refers to the stability of vehicle associations and message streams within an RSU's service region, not to movement of the RSU. Short association durations, unsuccessful transmissions, and locally divergent optimization directions may reduce the timeliness or usefulness of an RSU update. In this study, mobility stability is treated as an operational hypothesis rather than a causal claim. Longer association duration or smoother trajectories do not necessarily imply a more informative or less biased update. Its usefulness is therefore tested separately across the `_0709`, `_1416`, and `MixAll_0024` traffic regimes. The primary communication signal is derived from regional BSM reception because VeReMi Extension does not contain federated-update transport traces, but an additional FL-layer sensitivity experiment computes reliability directly from realized model-update deliveries under IID, heterogeneous, and burst-loss schedules. Update consistency is likewise not treated as a trust guarantee; coordinated one-, two-, and three-client corruption experiments explicitly test whether mutually aligned adversarial clients can bias the leave-one-client-out reference. Reliability-aware weighting also introduces a potential bias. Legitimate clients with small datasets, rare attack classes, skewed distributions, or distinctive mobility patterns may receive systematically lower weights. A credible aggregation method must therefore demonstrate both a performance benefit and the absence of detectable suppression under the evaluated conditions. Likewise, geographic partitioning alone does not establish a challenging non-IID setting. Client heterogeneity should be quantified using sample quantity, unique vehicles, complete class distributions, archive mixture, temporal coverage, and feature shift, together with explicit divergence measures.

Computational efficiency and interpretability impose additional requirements. Self-attention can represent dependencies across vehicular message sequences without recurrent state propagation, but standard Transformers may impose excessive parameter, storage, and latency costs on resource-constrained devices. This study therefore develops the *OBU-oriented Temporal Feature Transformer* (OBU-TFT), a compact model for short Basic Safety Message (BSM) sequences. This deployment-oriented motivation is consistent with recent IoV research that highlights federated/split learning and TinyML as complementary directions for reducing raw-data centralization while coping with resource-constrained edge devices (Aggarwal et al., 2025). Here, "TFT" denotes *Temporal Feature Transformer*; it does not refer to the canonical Temporal Fusion Transformer architecture. OBU-TFT uses two shallow sequential self-attention blocks and compact contextual fusion to refine short-term kinematic dependencies. SHAP is invoked only for diagnostic or post-event interpretation, and its computational cost is evaluated separately from the latency-critical detector.

To address these requirements, this study proposes *MoFed-Guard*, a reliability-aware federated framework for multiclass VANET misbehavior detection. The contribution is positioned as an empirically evaluated systems-level aggregation framework rather than as a new convergence-optimal or Byzantine-resilient optimization theory. Its central question is whether operational reliability signals can provide useful weighting information under controlled vehicular heterogeneity while their failure modes are made explicit. Fixed RSUs retain their locally observed BSM windows and train OBU-TFT, while vehicle OBUs use the resulting global model for local inference. The coordinator assigns weights to each RSU update using three operational indicators. These indicators capture vehicle-association stability, communication reliability, and consistency with the round-wise optimization direction. These indicators are not treated as direct evidence of client honesty; their association with update utility and client characteristics is evaluated empirically. The study addresses four research questions: (1) Does MoFed-Guard improve OBU-TFT when the local architecture, client partitions, training schedule, and random seeds are fixed? (2) Do the three reliability indicators provide complementary information without systematically

disadvantaging small, rare-class-rich, distributionally distinctive, or mobility-distinctive clients? (3) How does the complete framework compare with federated optimizers for non-IID data and client drift and with recurrent and Transformer-based detectors? (4) Does the framework retain multiclass performance, computational efficiency, and behaviorally meaningful explanations under the evaluated simulation conditions? These questions are examined through matched aggregation comparisons, component ablations, client-level heterogeneity analysis, repeated runs, class-wise evaluation, stress tests, efficiency measurements, and on-demand SHAP analysis. The interpretation distinguishes the simulated K-means RSU partition from measured roadside coverage while testing partition sensitivity with a road-network-aware alternative; it separates the BSM-derived primary communication signal from explicit FL-layer update-delivery simulations; and it evaluates both single-client and coordinated multi-client corruption to delimit the failure modes of cosine-based update consistency. Figure 1 illustrates the architecture and information flow. Vehicles exchange BSM/CAM messages through vehicle-to-vehicle and vehicle-to-infrastructure links. Fixed RSUs construct local temporal windows and act as federated training clients. Raw windows remain at the corresponding RSU; only model updates and lightweight reliability indicators are transmitted to the coordinator. MoFed-Guard calculates the aggregation weights and redistributes the updated global model for local OBU inference. SHAP explanations are generated only when diagnostic or post-event interpretation is requested and are excluded from the latency-critical path.

The principal contributions are as follows:

- This study introduces a data-local VANET misbehavior detection framework in which fixed RSUs act as federated training clients and vehicle OBUs execute the resulting compact 20-class detector. This design distinguishes infrastructure-assisted training from in-vehicle inference and avoids treating mobile vehicles and fixed RSUs as interchangeable federated participants.
- This research study develops *MoFed-Guard*, an empirically motivated reliability-aware aggregation mechanism that weights local updates according to vehicle-association stability, communication reliability, and leave-one-client-out update consistency. Its independent contribution is evaluated through matched aggregation baselines, component-removal ablations, utility correlations, client-bias auditing, explicit FL-layer update-delivery sensitivity, traffic-regime mobility analysis, and coordinated one-, two-, and three-client corruption tests. The method is not presented as a convergence-theoretic or Byzantine-robust aggregation principle.
- This work proposes *OBU-TFT*, an OBU-oriented Temporal Feature Transformer for short temporal BSM sequences containing eight kinematic channels and four derived contextual descriptors. In addition to single-block and parallel internal ablations, OBU-TFT is evaluated against parameter-matched Linformer and Performer implementations using macro-F1, parameter count, storage, and GPU/CPU latency. The resulting claim is explicitly Pareto-based.
- This study establishes a comprehensive non-IID evaluation protocol that characterizes simulated RSU clients according to sample quantity, unique vehicles, class composition, archive mixture, temporal distribution, and feature shift. Dependence on K-means geometry is tested with a road-network-aware RSU construction that introduces handover-like reassignment, while a reliability-bias audit examines whether the weighting mechanism systematically disadvantages small, rare-class-rich, distributionally distinctive, or mobility-distinctive clients.
- This research provides a multidimensional evaluation using macro-, weighted-, and class-level metrics. It also reports trajectory-clustered uncertainty and five-seed convergence behavior. Held-out client-scale tests are included to assess generalization across different numbers of federated clients. The evaluation further compares MoFed-Guard with strong federated aggregation baselines. Component ablations and loss-function ablations are also conducted. Additional sensitivity analyses examine FL transport, coordinated corruption, and spatial partitioning. Computational and communication costs are measured separately. On-demand SHAP explanations are used to support model interpretation.

The remainder of this article is organized as follows. Section 2 reviews related work on VANET misbehavior detection, federated learning, temporal modeling, and explainability. Section 3 presents the proposed methodology, including data preparation, trajectory-disjoint partitioning, OBU-TFT, MoFed-Guard aggregation, experimental protocols, and statistical analysis. Section 4 reports the predictive, class-wise, ablation, robustness, efficiency, and explainability results. Section 5 discusses the principal findings, their relationship to existing research, practical implications, and limitations. Finally, Section 6 concludes the study and outlines directions for future work.

2 Related work

2.1 Learning-based VANET misbehavior detection

Cryptographic mechanisms provide authentication, integrity, and nonrepudiation, but they cannot establish whether an authenticated message truthfully represents the sender's physical state. A malicious or faulty vehicle may therefore report false position, speed, acceleration, heading, or timing information while satisfying cryptographic checks. Learning-based misbehavior detection complements these mechanisms by identifying anomalous behavior from communication and mobility patterns (Alladi et al., 2023; Aloraini et al., 2024; Tyagi et al., 2024; Tariq and Ahanger, 2025). Conventional machine-learning methods have achieved strong results for position-falsification detection using message-level, plausibility-derived, and signal-based features (Rashid et al., 2023; Abdelkreem et al., 2026; Ercan et al., 2021). Abdelkreem et al. (2026) for example, combined a received-signal-strength-derived feature with a random-forest classifier and reported 99.86% accuracy and a 99.85% F1-score for binary classification of five position-falsification attacks on the original VeReMi dataset. These results demonstrate the value

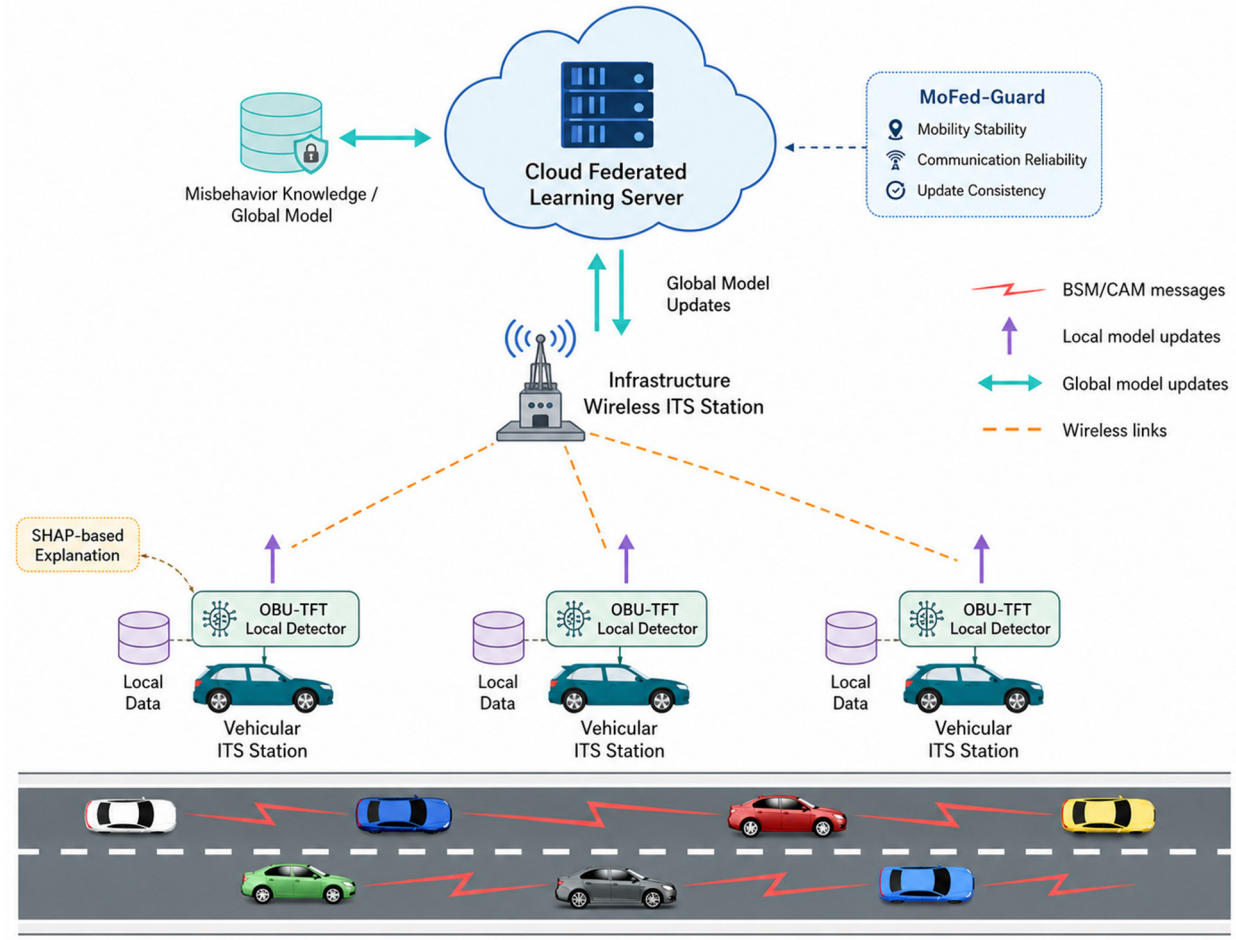


FIGURE 1
Architecture and information flow of the MoFed-Guard framework.

of domain-specific signal features; however, the binary objective, random test sampling, and centralized training protocol are not directly comparable with trajectory-disjoint, 20-class federated evaluation. Recent studies have expanded the design space through region-specific datasets, hybrid detectors, large-language-model representations, incremental adaptation, neighbor-assisted false-data checks, local hybrid reasoning, and RSU-side rule–learning integration (Bodkhe and Tanwar, 2025; Costa et al., 2024; Hamhoum and Cherkaoui, 2024; Nguyen et al., 2024; Idris et al., 2024; Amanullah et al., 2024; Park et al., 2024). VeReMi Extension broadens the original benchmark by adding attack variants, sensor-error models, traffic conditions, and message records (Van Der Heijden et al., 2018). VeMisNet applied engineered spatiotemporal and communication features with recurrent architectures to the benchmark's 20 categories (Youness et al., 2025). Its engineered-feature BiLSTM achieved 99.81% accuracy and a 99.81% F1-score on 500,000 samples, with 41.76 ms reported inference latency. This is a strong centralized benchmark, but its feature engineering, class balancing, sequence construction, data split, and computing protocol differ from those used here. Accordingly, cross-study values provide context rather than direct performance rankings. Because class imbalance can conceal weak recognition of infrequent attacks, multiclass evaluations should report macro-averaged and class-wise metrics in addition to overall accuracy.

2.2 Federated learning under vehicular heterogeneity

Federated learning (FL) enables distributed participants to optimize a shared model while retaining raw observations locally (Singh et al., 2025; Campos et al., 2024; Gurjar et al., 2025). Recent IoV literature also considers federated split learning and TinyML as complementary mechanisms for distributing computation across vehicles and edge infrastructure while limiting raw-data transfer and device burden (Aggarwal et al., 2025). Data locality reduces centralized collection, but it does not prevent reconstruction, inference, poisoning, or Byzantine attacks on model updates. Without evaluated protections such as secure aggregation or differential privacy, an FL system should therefore be described as *data-local*, not as providing formal privacy guarantees. Campos et al. (2024) applied FL to multiclass

misbehavior detection on the original VeReMi dataset using an optimized multilayer perceptron and SMOTE–Tomek rebalancing, reporting 93% overall accuracy. Gurjar et al. (2025) evaluated RNN- and LSTM-based classifiers on VeReMi Extension across several client counts using FedAvg, FedProx, and FedYogi. Although these studies establish the feasibility of collaborative vehicular detection, their client construction, observation selection, label mapping, partitioning, and metric definitions differ from the fixed-RSU, trajectory-disjoint protocol adopted here.

Unsupervised FL provides a complementary direction. Campos et al. (2025) combined a variational autoencoder, Gaussian-mixture modeling, restricted Boltzmann machine pretraining, and Fed+ aggregation, treating individual vehicles as clients on the original VeReMi dataset. For the analyzed 298-component client group, the complete model achieved 82.4% accuracy, 67.2% precision, 96.8% recall, and a 77.5% F1-score. The method reduces reliance on labeled attacks and accommodates vehicle-level non-IID data, but it performs anomaly discrimination rather than supervised identification of 20 classes. Other work has combined FL with LSTM-based in-vehicle intrusion detection, further illustrating the breadth of federated vehicular-security formulations (Miriam et al., 2025). FedAvg remains a useful reference, but sample-count weighting can be sensitive to client drift and non-IID data. More generally, VANET routing and QoS studies show that mobility, link dynamics, scalability, and rapidly changing topology can alter the communication conditions under which distributed applications operate (Chauhan et al., 2016; Radaur, 2015; Rasheed et al., 2017). These networking results motivate communication-aware learning, but they do not justify equating packet-level or BSM reception behavior with the success of an FL model-update transfer. Proximal regularization, adaptive server optimization, and alternative aggregation rules can improve optimization without explicitly representing mobility or communication reliability. In vehicular FL, a divergent update may reflect unstable optimization, a rare class, a small local dataset, or a distinct mobility region. Reliability-aware weighting must therefore be audited against sample count, class coverage, distributional divergence, mobility, and update utility to ensure that operationally important clients are not systematically suppressed.

2.3 Transformer-based temporal detection and deployment scope

Recurrent, hybrid, and Transformer-based architectures have been investigated to improve temporal and contextual modeling of V2X behavior (Costa et al., 2024; Li et al., 2025; Kalogiannis et al., 2026; Missara et al., 2025). Self-attention can model relationships among temporally separated messages without relying exclusively on recurrent state propagation. AttentionGuard applies a multi-head Transformer encoder to sequential kinematic observations in vehicular platoons and reports an F1-score of up to 0.95 across constant, gradual, and combined falsification attacks, with 100 ms decision intervals (Li et al., 2025). Its evaluation supports attention-based detection during steady-state, joining, and exiting maneuvers, but its platoon-specific attack-detection task differs from general 20-class VeReMi Extension classification. Missara et al. (2025) developed a Transformer-based detector for V2X 5G edge networks and evaluated it on VeReMi Extension. The model achieved 99.37% accuracy, 98.69% recall, and a 99.30% F1-score. These findings support the representational capacity of Transformers for vehicular behavior; however, the edge-assisted architecture, preprocessing, partitioning, and computing environment differ from the data-local RSU training and compact OBU inference considered here. Standard Transformer configurations may also impose substantial memory and latency costs on resource-constrained vehicular hardware. Efficient-attention architectures such as Linformer and Performer reduce the computational burden of conventional self-attention through low-rank projection and kernel-based approximation, respectively (Wang et al., 2020; Choromanski et al., 2020); these models therefore provide relevant parameter-matched comparators for the deployment-oriented OBU-TFT design.

The local model proposed in this study, OBU-TFT, denotes an *OBU-oriented Temporal Feature Transformer*; it is not an implementation of the canonical Temporal Fusion Transformer. The architectural claim is evaluated explicitly as an accuracy–parameter–latency trade-off. OBU-TFT is compared not only with single-encoder and parallel-encoder internal variants but also with parameter-matched Linformer and Performer implementations. In the matched experiments, OBU-TFT achieved 94.52% macro-F1 with 85,460 parameters and 5.10 ms T4 latency, compared with 94.08%/82,944/4.84 ms for Linformer and 94.24%/88,320/5.36 ms for Performer. Thus, OBU-TFT provides the highest predictive score, while Linformer remains smaller and faster; both occupy non-dominated points under the predefined Pareto criterion. Because latency is measured in a controlled benchmark rather than on production automotive hardware, the model is characterized as lightweight and deployment-oriented, not field-validated.

2.4 Explainability and research gap

Explainability supports model debugging, feature validation, and post-event analysis in safety-critical vehicular systems. Shapley additive explanations (SHAP) identify variables associated with a prediction, but they do not establish malicious intent or causal responsibility. Explanation generation also adds computational cost. MoFed-Guard therefore uses SHAP as an on-demand diagnostic layer outside the latency-critical classification path; reported detector latency excludes explanation generation. Table 1 positions MoFed-Guard against seven representative studies. Several centralized and edge-assisted methods report accuracies above 99% (Abdelkreem et al., 2026; Youness et al., 2025; Missara et al., 2025), but those results arise from different objectives, features, splits, balancing procedures, or infrastructure assumptions. Existing federated studies demonstrate data-local collaborative training (Campos et al., 2024; Gurjar et al., 2025; Campos et al., 2025), but they do not jointly evaluate fixed-RSU clients, trajectory-disjoint 20-class classification, reliability weighting based on vehicle-association stability, communication reliability and update consistency, compact Transformer inference, reliability-bias auditing, repeated runs, and on-demand explanations. The present work frames MoFed-Guard as

TABLE 1 Quantitative and protocol-aware comparison with representative VANET misbehavior detection studies.

Study	Dataset and task	Learning configuration	Reported result	Scope relative to the present study
Abdelkreem et al. (2026)	Original VeReMi; binary combined position-falsification detection	Centralized RF with derived RSSI-based features	99.86% accuracy and 99.85% F1-score	Strong binary performance; different label space, random test sampling, engineered signal features, and no federated training
Campos et al. (2024)	Original VeReMi; multiclass misbehavior detection	Federated optimized MLP with SMOTE–Tomek balancing	93% overall accuracy	Demonstrates data-local collaborative detection but does not jointly model mobility, communication reliability, and update consistency
Gurjar et al. (2025)	VeReMi Extension; federated misbehavior classification	RNN/LSTM variants with FedAvg, FedProx, and FedYogi across client counts	Performance evaluated across multiple client and aggregation configurations	Closest federated conceptual comparator; client construction, partitioning, labels, and metric definitions differ from the present protocol
Li et al. (2025)	Simulated vehicular platoons; falsification-attack detection	Multi-head Transformer encoder using sequential kinematic data	Up to 0.95 F1-score; 100 ms decision intervals	Platoon-specific task with different attacks and inputs; does not evaluate general 20-class VeReMi Extension detection or FL aggregation
VeMisNet (Youness et al., 2025)	VeReMi Extension; centralized classification of 20 categories	Engineered spatiotemporal features with BiLSTM	99.81% accuracy and F1-score on 500,000 samples; 41.76 ms latency	Strong centralized benchmark; different balancing, feature engineering, split, sample scale, and computing protocol
Missara et al. (2025)	VeReMi Extension; V2X misbehavior detection	Transformer-based detection in a 5G edge setting	99.37% accuracy, 98.69% recall, and 99.30% F1-score	Edge-assisted architecture and different preprocessing, partitioning, and hardware protocol limit direct comparison
Campos et al. (2025)	Original VeReMi; unsupervised anomaly-oriented detection	Vehicle clients with VAE, GMM, RBM pretraining, and Fed+	82.4% accuracy, 67.2% precision, 96.8% recall, and 77.5% F1-score	Realistic vehicle-level non-IID formulation, but binary anomaly detection does not provide supervised 20-class discrimination
MoFed-Guard	**VeReMi Extension; trajectory-disjoint 20-class classification**	**Fixed RSU clients, OBU-TFT inference, and reliability-aware aggregation**	**95.20% accuracy, 94.75% macro-precision, 94.30% macro-recall, and 94.52% macro-F1; 5.10 ms latency**	**Integrates compact temporal detection, mobility- and communication-aware aggregation, repeated-run evaluation, reliability-bias auditing, and on-demand SHAP explanations**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

a controlled empirical integration of these elements. The principal validity concerns are evaluated directly. Mobility contribution is tested across three traffic regimes, while communication reliability is re-evaluated using explicit FL-layer delivery schedules. Spatial dependence is examined through a road-network-aware RSU construction, and parameter-matched efficient-attention baselines are included. Coordinated-corruption experiments further show that cosine-based consistency can fail when multiple adversarial clients are jointly aligned.

The contribution of MoFed-Guard is therefore not defined as a fundamentally new aggregation theorem or solely by aggregate accuracy. Instead, it is evaluated as a systems-level combination of reliability-conditioned federated optimization, compact temporal detection, explicit heterogeneity characterization, assumption-level sensitivity testing, and *post hoc* auditing under a controlled simulation protocol. It addresses multiclass detection, data-local federated optimization, mobile-client reliability, computational compactness, and interpretability within one controlled protocol. Direct performance differences are calculated only for baselines reproduced with identical partitions, client assignments, random seeds, training schedules, metric definitions, and computing environments. This restriction prevents unsupported conclusions from incompatible cross-study comparisons.

3 Proposed methodology

This section presents the controlled, simulation-based protocol used to evaluate MoFed-Guard for 20-class VANET misbehavior detection. The workflow comprises dataset curation, trajectory-disjoint partitioning, temporal-window construction, RSU-based federated training, reliability-aware aggregation, held-out evaluation, and on-demand SHAP interpretation. Fixed roadside units (RSUs) act as federated clients, whereas vehicle on-board units (OBUs) execute the trained detector.

Raw BSM windows remain in their RSU partitions; only model updates and scalar reliability indicators are transmitted to the coordinator. The experiments emulate vehicular operation and do not constitute validation on a production wireless network or automotive platform.

3.1 Study design, research questions, and controls

The primary experiment compares MoFed-Guard with FedAvg while holding the OBU-TFT local model, partitions, initialization seeds, local epochs, optimizer, client schedules, and stopping criteria constant. This matched design isolates the aggregation effect. Architecture comparisons evaluate RNN-LSTM, BiLSTM, Vanilla Transformer, Transformer-XL, and Reformer under FedAvg. Aggregation comparisons evaluate OBU-TFT with FedProx, SCAFFOLD, FedAdam, FedYogi, coordinate-wise trimmed mean, and Krum when the required client-count assumptions are satisfied (Li et al., 2020; Karimireddy et al., 2020; Reddi et al., 2020; Blanchard et al., 2017; Yin et al., 2018). The experiments address four questions: (i) Does MoFed-Guard improve OBU-TFT under matched conditions? (ii) Do mobility stability, communication reliability, and update consistency provide complementary information across materially different traffic and update-delivery conditions? (iii) Do the resulting weights suppress clients with rare classes, small datasets, unusual label distributions, or distinctive mobility patterns, and how do they behave when several clients submit mutually aligned but incorrect updates? (iv) Does the framework retain class-wise performance and computational efficiency as RSU count, spatial client construction, communication reliability, and local attention architecture change? To strengthen construct validity and robustness assessment, the protocol includes five targeted sensitivity analyses: traffic-regime mobility analysis, FL-layer update-delivery simulation independent of BSM reception, coordinated multi-client corruption, an alternative road-network-aware RSU association scheme, and parameter-matched lightweight attention baselines. These experiments test the scope and failure modes of the proposed signals without treating them as causal or Byzantine-robust trust guarantees.

The principal configurations contain 3, 5, 8, and 10 RSU clients. The primary experiment uses full scheduled participation. Stress tests emulate 20% and 40% random client unavailability, controlled FL-update delivery failures, and one to three local epochs. The principal clean architecture and aggregation comparisons are repeated with seeds 11, 23, 37, 42, and 59. Component, encoder, loss, client-scale, availability, transport-sensitivity, partition-sensitivity, and adversarial analyses use the fixed diagnostic seed 42 unless explicitly reported across the five matched seeds. The primary outcome is held-out macro-F1. Secondary outcomes are accuracy; macro-, micro-, weighted, and class-wise precision, recall, and F1; balanced accuracy; faulty-behavior and attack-group performance; convergence; client-scale sensitivity; model size; latency; round time; communication cost; aggregation overhead; SHAP cost; reliability-weight bias; and adversarial-client consistency.

The corruption protocol is extended from a single-client test to $f \in \{1, 2, 3\}$ corrupted clients in the 10-client configuration. The same adversarial-client identities and round schedules are reused across aggregation methods. Two coordinated attacks are evaluated. In coordinated sign flip, every adversarial client returns $-\Delta\theta_k^{(t)}$. In the aligned-bias diagnostic, all adversarial clients return a common direction opposite to the mean benign update, scaled to the median benign update norm. Let $\mathcal{B}_t$ denote the non-adversarial clients, $\bar{\Delta}_{\mathcal{B}}^{(t)} = |\mathcal{B}_t|^{-1} \sum_{j \in \mathcal{B}_t} \Delta\theta_j^{(t)}$, and $m_t = \text{median}_{j \in \mathcal{B}_t} \|\Delta\theta_j^{(t)}\|_2$. The aligned adversarial update is

$$\widetilde{\Delta\theta}_k^{(t)} = -m_t \frac{\bar{\Delta}_{\mathcal{B}}^{(t)}}{\|\bar{\Delta}_{\mathcal{B}}^{(t)}\|_2 + \varepsilon}, \qquad k \in \mathcal{A}_t.$$

This deliberately strong diagnostic creates mutually consistent but collectively incorrect updates and therefore directly probes the failure mode of cosine consistency against a leave-one-client-out mean. It is used to characterize vulnerability, not to claim an attacker has access to the benign mean in deployment.

3.2 VeReMi extension dataset and label space

The experiments use VeReMi Extension, which expands VeReMi with additional attacks, sensor-error models, and traffic conditions (Van Der Heijden et al., 2018; Kamel et al., 2020). The underlying simulations were generated with OMNeT++, VEINS, and the Luxembourg SUMO Traffic (LuST) scenario. The prepared release comprises 39 archives. It includes 19 attack-specific high-density archives with the suffix `_0709` and 19 corresponding low-density archives with the suffix `_1416`. The full-day mixed archive, `MixAll_0024`, is also included. The suffixes represent the 07:00–09:00 and 14:00–16:00 intervals; `MixAll_0024` covers 00:00–24:00. The task retains 20 mutually exclusive labels: one normal class, eight faulty-behavior classes, and eleven malicious-attack classes. These groupings support secondary analysis only; training and evaluation use the original 20-class labels. For the weighting audit, a rare class is operationally defined as a class in the lowest quartile of window support in the training partition. Membership is fixed before federated training and is not recalculated by client or test partition. Each retained JSON observation contains a simulation identifier, sender identifier, transmission time, ground-truth label, and eight kinematic values. The source receiver log supplies the receiver identifier used only to estimate regional BSM reception; it is not a classifier input. The temporal feature order is

$$\mathbf{x}_t = [\texttt{posx}, \texttt{posy}, \texttt{aclx}, \texttt{acly}, \texttt{spdx}, \texttt{spdy}, \texttt{hedx}, \texttt{hedy}]_t. \tag{1}$$

The sender identifier, simulation identifier, and transmission time are used only to reconstruct trajectories, prevent leakage, establish temporal order, and assign trajectories to RSU regions. They are not supplied to the classifier. Table 2 reports the global message-level distribution before temporal-window construction.

TABLE 2 Global class distribution of the VeReMi extension data used in this study.

Type	Class	Group	Messages
0	Normal	Normal	1,900,539
1	Constant position	Faulty	43,653
2	Constant position offset	Faulty	43,567
3	Random position	Faulty	43,857
4	Random position offset	Faulty	42,575
5	DoS	Attack	41,925
6	DoS random	Attack	44,359
7	DoS random sybil	Attack	42,258
8	Data replay	Attack	42,583
9	Data replay sybil	Attack	42,790
10	Disruptive	Attack	43,264
11	DoS disruptive sybil	Attack	44,337
12	DoS disruptive	Attack	43,118
13	Eventual stop	Attack	131,305
14	Grid sybil	Attack	126,724
15	Constant speed	Faulty	129,270
16	Constant speed offset	Faulty	175,391
17	Delayed messages	Attack	44,310
18	Random speed	Faulty	86,883
19	Random speed offset	Faulty	82,100

3.3 Leakage-resistant splitting and preprocessing

Malformed observations, exact duplicates, and records missing an essential kinematic field are removed. Records are grouped by simulation run and physical trajectory using the simulation identifier and persistent ground-truth vehicle identifier; sender pseudonyms are linked when the metadata support unambiguous linkage. A trajectory containing a label transition is segmented at the transition before partitioning, so no model window contains multiple labels. Entire labeled trajectory segments, rather than messages or windows, are assigned to training, validation, or test data. Stratified group allocation produces approximate 70/15/15% partitions while preserving label and traffic-scenario representation. No physical trajectory or sender group occurs in more than one partition. Temporal windows are constructed after the trajectory-level split. Messages within each labeled trajectory segment are ordered chronologically and divided into non-overlapping windows of length $T = 20$ with a stride of 20; incomplete terminal windows are discarded. Each window inherits the segment label. Thus, messages are not reused, windows do not cross label transitions or subset boundaries, and neighboring observations from the same physical trajectory cannot occur in different partitions.

The four physical context descriptors are first computed from unscaled kinematic values. The eight temporal channels and four context descriptors are then scaled independently to $[0, 1]$ using parameters estimated only from training trajectories. For any temporal or context feature j,

$$x'_{t,j} = \frac{x_{t,j} - x^{\text{train}}_{j,\min}}{x^{\text{train}}_{j,\max} - x^{\text{train}}_{j,\min} + \varepsilon}, \qquad \varepsilon = 10^{-8}. \tag{2}$$

The frozen training parameters are applied to validation and test data, with out-of-range values clipped to $[0, 1]$. No resampling, synthetic example generation, or scaling parameter is estimated from validation or test data.

For trajectory segment i, the unscaled temporal window is

$$X_i^{\text{raw}} = [\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,20}]^\top \in \mathbb{R}^{20\times 8}. \tag{3}$$

The contextual descriptors are deterministic functions of X_i^{raw}, not additional message fields. Speed magnitude and heading angle are

$$s_{i,t} = \sqrt{\text{spdx}_{i,t}^2 + \text{spdy}_{i,t}^2}, \qquad h_{i,t} = \text{atan2}(\text{hedy}_{i,t}, \text{hedx}_{i,t}). \tag{4}$$

Equations 2–4 describe the normalization, temporal-window representation, and extraction of speed and heading information, respectively. The unscaled contextual vector is

$$C_i^{\text{raw}} = [\text{Var}(s_i), \text{CVar}(h_i), \|\mathbf{p}_{i,20} - \mathbf{p}_{i,1}\|_2, |\,\text{wrap}(h_{i,20} - h_{i,1})|]^\top \in \mathbb{R}^4, \tag{5}$$

where CVar is circular variance and wrap maps angular differences to $[-\pi, \pi]$. Training-only scaling produces the model inputs X_i and C_i. Neither `sender`, receiver identity, nor `sendTime` is a predictor.

3.4 Traffic-Regime mobility sensitivity protocol

The mobility component is evaluated separately in the `_0709`, `_1416`, and `MixAll_0024` archive regimes to determine whether its empirical contribution persists across materially different observed traffic conditions. These regimes are not assumed to differ only in mobility; before model comparison, each is characterized by mean speed, speed variability, stopped-observation ratio, heading-change rate, association duration, trajectory count, class composition, and feature shift. Trajectory-disjoint training, validation, and test allocation is preserved within each regime.

For each regime g, two quantities are reported. First, the association between mobility stability and one-client validation utility is measured using Spearman's $\rho_g(S, U)$. Second, the full MoFed-Guard model is compared with the otherwise identical "without mobility stability" ablation on the held-out regime-specific test partition. The difference

$$\Delta F_{1,g}^{(S)} = F_{1,g}^{\text{full}} - F_{1,g}^{(-S)} \tag{6}$$

quantifies whether the mobility term contributes positively under each observed traffic regime. A consistent positive $\Delta F_{1,g}^{(S)}$ across

regimes is interpreted as cross-regime empirical support, not causal proof that mobility stability determines gradient quality, because archive regime also changes traffic composition and other covariates.

3.5 RSU-client construction and non-IID characterization

Training trajectories are assigned to federated clients. For each $K \in \{3, 5, 8, 10\}$, K-means clustering with Euclidean distance is fitted to the median position of every training trajectory using k-means++ initialization, 50 restarts, and the matched experimental seed. Each complete trajectory is assigned to its nearest center. The centers represent simulated fixed-RSU service regions, not measured RSU locations. A trajectory is never divided among clients, and validation and test trajectories do not influence center estimation. Changing K therefore repartitions the same training set; it does not add geographic data or alter the validation and test sets. This K-means construction is an experimental partitioning device rather than a propagation-calibrated RSU deployment model. It does not reproduce radio attenuation, interference, or geographically correlated link failures; consequently, the primary clients are interpreted as controlled spatial service regions rather than a faithful roadside deployment.

To test whether the main aggregation result depends on K-means partitioning, a second road-network-aware association protocol is evaluated as a sensitivity analysis using the LuST SUMO road graph corresponding to the VeReMi simulation. For each tested K, K fixed RSU anchors are selected from traversed road junctions using deterministic farthest-point sampling initialized at the junction nearest the geometric center of the occupied training-road network. This selection uses no class labels and no validation or test observations. Each training window is associated with the nearest anchor by road-network distance using its median physical position. Consequently, a physical trajectory may change RSU association across successive non-overlapping windows when the nearest road-network anchor changes, thereby introducing an explicit handover-like reassignment absent from the trajectory-level K-means construction. The trajectory-disjoint train/validation/test split is performed before this association step, so the sensitivity analysis does not create cross-partition leakage.

The primary matched FedAvg–MoFed-Guard comparison is repeated under this road-network-aware client construction for the same client counts for which all clients contain sufficient training windows. Agreement in the direction of the aggregation effect across K-means and road-network-aware associations is treated as evidence that the result is not specific to one spatial partitioning rule. This experiment still does not model physical-layer propagation, interference, or measured V2I handover signaling and is therefore a partition-robustness test rather than a field-calibrated radio evaluation.

For each RSU client k, message-level records characterize partition composition before windowing, whereas n_k denotes the number of complete model windows used for local optimization. The analysis reports unique trajectories and vehicles; the 20-class message-count vector; normal, faulty, and attack totals; source-archive proportions; temporal coverage; rare-class coverage; and feature means and standard deviations. Binary misbehavior totals are supplementary and do not define the learning task.

Label heterogeneity is quantified by Jensen–Shannon divergence between the client class distribution p_k and global training distribution p:

$$D_{\mathrm{JS}}(p_k \| p) = \frac{1}{2} D_{\mathrm{KL}}(p_k \| m_k) + \frac{1}{2} D_{\mathrm{KL}}(p \| m_k), \qquad m_k = \frac{p_k + p}{2}. \tag{7}$$

Temporal heterogeneity is measured by the Jensen–Shannon divergence between client and global hourly histograms. Feature heterogeneity is measured as the mean normalized one-dimensional Wasserstein distance across the eight temporal channels:

$$D_{\mathrm{feat},k} = \frac{1}{8} \sum_{j=1}^{8} \frac{W_1(F_{k,j}, F_j)}{\sigma_j^{\mathrm{train}} + \varepsilon}, \tag{8}$$

where $F_{k,j}$ and F_j are the empirical client and global training distributions of feature j. Client-size imbalance is summarized by $\max_k n_k / \min_k n_k$ and the coefficient of variation of $\{n_k\}$. These measures distinguish label skew, feature skew, temporal skew, and quantity skew instead of treating geographic allocation alone as evidence of non-IID data. Equations 7, 8 quantify client label heterogeneity and feature heterogeneity, respectively.

3.6 OBU-TFT input representation and architecture

OBU-TFT denotes the *OBU-oriented Temporal Feature Transformer*; it is not the canonical Temporal Fusion Transformer (Lim et al., 2021). It omits static-covariate encoders, variable-selection networks, recurrent local-processing layers, gated residual networks, and forecasting heads. The name instead refers to compact temporal-feature projection and attention-based classification for OBU-oriented inference. Figure 2 summarizes the workflow. Fixed RSUs perform local training, while the coordinator aggregates client updates. Vehicle OBUs execute the trained detector. SHAP is applied only to selected diagnostic cases outside the latency-critical inference path.

The eight-channel window is projected to a 64-dimensional latent space and combined with positional encoding:

$$Z_i = X_i W_e + b_e + \mathrm{PE}, \tag{9}$$

where $W_e \in \mathbb{R}^{8 \times 64}$. Within an encoder block,

$$Q = ZW_Q, \qquad K = ZW_K, \qquad V = ZW_V, \tag{10}$$

and

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{11}$$

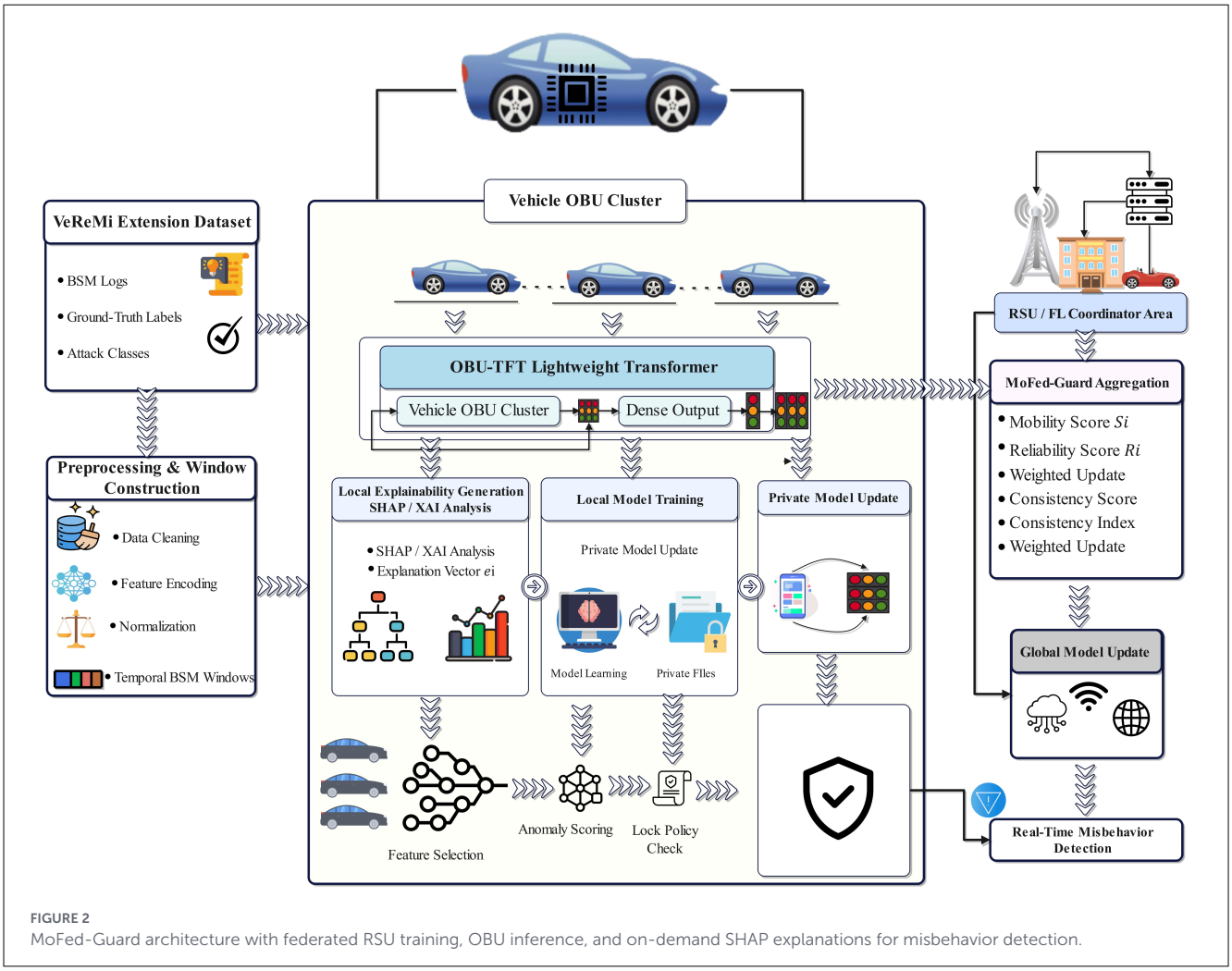


FIGURE 2
MoFed-Guard architecture with federated RSU training, OBU inference, and on-demand SHAP explanations for misbehavior detection.

Four attention heads with $d_k = 16$ are concatenated and projected. Each of the two sequential encoder blocks contains multi-head attention, residual connections, one layer-normalization stage, dropout, and a feed-forward network of widths 128 and 64. This single-normalization configuration is the implemented lightweight block, not the standard two-normalization Transformer block. The second block receives the complete output of the first. Global average pooling produces

$$\bar{H}_i = \frac{1}{20}\sum_{t=1}^{20} H_{i,t}. \tag{12}$$

The contextual vector is projected to 64 dimensions, $\widetilde{C}_i = C_i W_c + b_c$, and fused without an additional learned gate matrix:

$$G_i = \sigma(\bar{H}_i + \widetilde{C}_i), \qquad F_i = G_i \odot \bar{H}_i + (1 - G_i) \odot \widetilde{C}_i. \tag{13}$$

The fused vector passes through dense layers of widths 128 and 64 and a 20-class softmax output:

$$\widehat{\mathbf{y}}_i = \mathrm{softmax}(W_o F_i + b_o), \qquad \widehat{c}_i = \arg\max_{c\in\{0,\ldots,19\}} \widehat{y}_{i,c}. \tag{14}$$

Equations 9–14 define the OBU-TFT embedding, attention operations, temporal pooling, gated fusion, and multiclass prediction. Table 3 specifies the output shape, parameter count, and input of each OBU-TFT layer. The trainable entries, including the context branch, sum to 85,460 parameters.

3.6.1 Encoder-arrangement ablation

Two alternatives test the sequential design. The single-encoder variant removes block 2 and contains 52,116 parameters. The parallel variant sends the same embedding to two independent blocks, concatenates their pooled 64-dimensional outputs, and projects the resulting 128-dimensional vector to 64 dimensions before context fusion; it contains 93,716 parameters. Inputs, optimizer, client partitions, seeds, and output head are matched. Macro-F1 is reported with parameter count, model size, and batch-one latency. The comparison is interpreted as an accuracy–complexity trade-off rather than evidence that the sequential design dominates all lightweight attention architectures. A configuration is considered preferable only when macro-F1, parameter count, storage, and latency are evaluated jointly.

TABLE 3 Layer-wise specification of the two-sequential-encoder OBU-TFT.

Layer	Output shape	Parameters	Input
Input BSM window	(20, 8)	0	—
Linear embedding	(20, 64)	576	BSM window
Positional encoding	(20, 64)	0	Embedding
Multi-head attention block 1	(20, 64)	16,640	Positional encoding
Feed-forward network 1	(20, 64)	16,576	Attention representation
Layer normalization 1	(20, 64)	128	Encoder residual output 1
Multi-head attention block 2	(20, 64)	16,640	Encoder block 1
Feed-forward network 2	(20, 64)	16,576	Attention representation
Layer normalization 2	(20, 64)	128	Encoder residual output 2
Context input and projection	(64)	320	Four derived descriptors
Global average pooling	(64)	0	Encoder block 2
Gated fusion	(64)	0	Temporal and context vectors
Dense layer	(128)	8,320	Fused representation
Dense layer	(64)	8,256	Dense layer
Softmax output	(20)	1,300	Dense layer
Total	—	**85,460**	—

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

Input: Ordered sender stream $\mathcal{B}_i$; fitted training scalers $\mathcal{S}_X$ and $\mathcal{S}_C$; trained model parameters θ
Output: Class-probability vector $\hat{\mathbf{y}}_i$ and predicted class $\hat{c}_i$

// Stage 1: Window construction and preprocessing
$X_i^{\text{raw}} \leftarrow \text{Window}(\mathcal{B}_i, T = 20, d = 8) \in \mathbb{R}^{20\times 8}$
Order the eight temporal channels according to Equation 1
Derive $C_i^{\text{raw}} \in \mathbb{R}^4$ from the physical-unit window according to Equation 5
$X_i \leftarrow \text{clip}_{[0,1]}\left(\mathcal{S}_X(X_i^{\text{raw}})\right)$
$C_i \leftarrow \text{clip}_{[0,1]}\left(\mathcal{S}_C(C_i^{\text{raw}})\right)$
// Stage 2: Temporal representation learning
$Z_i \leftarrow X_i W_e + b_e + \text{PE}$
$H_i^{(1)} \leftarrow \text{Encoder}_1(Z_i)$
$H_i \leftarrow \text{Encoder}_2(H_i^{(1)})$
$\bar{H}_i \leftarrow \frac{1}{20}\sum_{t=1}^{20} H_{i,t}$
// Stage 3: Context-aware gated fusion
$\widetilde{C}_i \leftarrow C_i W_c + b_c$
$G_i \leftarrow \sigma\left(\bar{H}_i + \widetilde{C}_i\right)$
$F_i \leftarrow G_i \odot \bar{H}_i + \left(\mathbf{1} - G_i\right) \odot \widetilde{C}_i$
// Stage 4: Multiclass misbehavior prediction
$D_i^{(1)} \leftarrow \phi(W_1 F_i + b_1) \in \mathbb{R}^{128}$
$D_i^{(2)} \leftarrow \phi(W_2 D_i^{(1)} + b_2) \in \mathbb{R}^{64}$
$\hat{\mathbf{y}}_i \leftarrow \text{softmax}(W_o D_i^{(2)} + b_o) \in \mathbb{R}^{20}$
$\hat{c}_i \leftarrow \arg\max_{c\in\{0,\ldots,19\}} \hat{y}_{i,c}$
Mark the processed observations as consumed to prevent window overlap
return $\hat{\mathbf{y}}_i, \hat{c}_i$

Algorithm 1. OBU-TFT local misbehavior inference.

3.6.2 Parameter-matched lightweight attention baselines

To determine whether OBU-TFT's performance gain is attributable only to a larger attention budget, two efficient-attention baselines are evaluated. Linformer applies a low-rank projection to the sequence dimension (Wang et al., 2020), while Performer approximates softmax attention using kernel-based random features (Choromanski et al., 2020). Both models use the same (20, 8) temporal input, the same four context descriptors, global pooling, 20-class output, optimizer, local epochs, client partitions, and early-stopping rule as OBU-TFT. Their embedding widths, feed-forward widths, and projection/random-feature dimensions are adjusted using training and validation data only so that each model lies within ±10% of OBU-TFT's 85,460-parameter budget. No test metric is used for parameter matching or architecture selection. Architecture comparison is performed with the aggregation rule held fixed so that local-model effects are not confounded with federation effects. Each model is evaluated using macro-F1, trainable parameters, 32-bit storage, batch-one T4 latency, and CPU latency. Pareto analysis uses the three objectives of maximizing macro-F1 while minimizing parameter count and latency. Model a dominates model b only if a is no worse on all three objectives and strictly better on at least one. This analysis permits OBU-TFT to be described as Pareto-efficient only if supported by the measured comparison; otherwise, the result is reported as an explicit accuracy–resource trade-off.

Algorithm 1 summarizes the complete local detection path, from sender-specific window construction and training-only normalization to temporal–context fusion and 20-class prediction. It also makes explicit that the OBU executes inference locally and does not participate as a federated training client.

3.7 Local objective and class-imbalance controls

The primary experiments use categorical cross-entropy without oversampling, thereby retaining the observed client distributions:

$$L_k(\theta) = -\frac{1}{n_k}\sum_{i=1}^{n_k}\sum_{c=0}^{19} y_{i,c}\log \widehat{y}_{i,c}. \tag{15}$$

MoFed-Guard is not presented as a class-imbalance solution because minority-class coverage is absent from its aggregation score. Under the same OBU-TFT and aggregation rule, two controlled alternatives are evaluated. Class-weighted cross-entropy, with $a_c = N/(20N_c)$ normalized to mean one, and focal loss with $\zeta = 2$. Class weights use the global training partition only. Loss sensitivity is summarized using hard-class recall, defined as macro-recall over classes 4, 7, 11, 12, and 17. These five classes had the lowest validation recall under unweighted cross-entropy and were fixed before test evaluation. Rare-class share in the weighting-bias audit remains a separate support-based measure defined by the lowest training-support quartile.

3.8 Federated optimization and baselines

Let $\mathcal{V}_t$ denote the RSUs participating in round t. Each client receives $\theta^{(t)}$, performs three Adam epochs, and returns

$$\Delta\theta_k^{(t)} = \theta_k^{(t+1)} - \theta^{(t)}. \tag{16}$$

The local learning rate is 5×10^{-4} and the batch size is 256. Training runs for at most 100 rounds. Early stopping monitors validation loss with patience 10 and a minimum improvement of 10^{-4}; the checkpoint with the highest validation macro-F1 among completed rounds is retained. The test set is accessed once per seed after model and coefficient selection. All aggregation baselines use OBU-TFT locally. FedProx selects $\mu \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ by validation macro-F1. SCAFFOLD uses client and server control variates with the matched learning rate and schedule. FedAdam and FedYogi use $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\tau = 10^{-3}$, and select the server learning rate from $\{10^{-3}, 10^{-2}, 10^{-1}\}$. In the clean comparison, coordinate-wise trimmed mean removes one largest and one smallest value per coordinate and Krum is configured for $f = 1$. In the coordinated-corruption sensitivity analysis, the robust baselines are configured with the known experimental corruption budget $f \in \{1, 2, 3\}$: trimmed mean removes the f largest and f smallest coordinate values when $N_t > 2f$, and Krum is evaluated only when $N_t \geq 2f + 3$. This prevents the robust baselines from being artificially disadvantaged by retaining an $f = 1$ setting during multi-client corruption. Each method is tuned independently on validation data. Equations 15, 16 specify the local cross-entropy objective and federated model update, respectively.

3.9 MoFed-Guard reliability-aware aggregation

MoFed-Guard uses three operational indicators. They are not evidence that a client is honest; Section 3.10 tests their relationships with update utility and minority-class representation. Because RSUs are fixed, mobility stability describes the vehicles observed within a service region, not RSU motion.

3.9.1 Mobility stability

Let $\mathcal{U}_k^{(t)}$ be the vehicles contributing local windows when RSU k participates in round t. The association summary is precomputed from physical trace time: $\tau_{k,u}^{(t)}$ is the elapsed time between the first and last received BSM of vehicle u assigned to region k, and $\kappa_{k,u}^{(t)}$ is trajectory continuity. The superscript t identifies the optimization round and participating local sample; it does not imply that an FL round is a physical-time interval. Equation 17 defines the trajectory-continuity component used to calculate mobility stability.

$$\kappa_{k,u}^{(t)} = \exp\left[-\frac{1}{T-2}\sum_{m=2}^{T-1}\frac{\|\Delta\mathbf{p}_u^{m+1} - \Delta\mathbf{p}_u^{m}\|_2}{d_0 + \varepsilon}\right], \tag{17}$$

where d_0 is the median displacement-change magnitude in the training partition. RSU-level mobility stability is

$$S_k^{(t)} = \frac{1}{|\mathcal{U}_k^{(t)}|}\sum_{u\in\mathcal{U}_k^{(t)}}\left[\lambda\frac{\tau_{k,u}^{(t)}}{\tau_{\max}^{(t)} + \varepsilon} + (1-\lambda)\kappa_{k,u}^{(t)}\right], \tag{18}$$

with $\lambda = 0.50$. Larger values indicate longer observed associations and smoother trajectories, not benign class membership. Because complete trajectories are assigned to their dominant region, this experiment does not model explicit inter-RSU handover. The metric is therefore an operational continuity descriptor, not a theoretical estimator of gradient quality. A long or smooth association can still yield a biased or uninformative local update, and the observed relationship between $S_k^{(t)}$ and update utility is treated as an empirical association rather than a causal guarantee. The present dataset also does not support a controlled mobility-regime intervention that would establish invariance of this relationship under substantially different road-topology or mobility processes.

3.9.2 Communication reliability

Communication reliability is computed over a trailing window of $L = 10$ rounds:

$$R_k^{(t)} = \frac{\sum_{\ell=\max(1,t-L+1)}^{t} n_{k,\text{succ}}^{(\ell)} + \varepsilon}{\sum_{\ell=\max(1,t-L+1)}^{t} n_{k,\text{att}}^{(\ell)} + \varepsilon}. \tag{19}$$

Here, $n_{k,\text{att}}^{(\ell)}$ counts scheduled client–server model updates and $n_{k,\text{succ}}^{(\ell)}$ counts model updates delivered before the simulated

deadline. VeReMi Extension contains no federated-update transport trace. To test the BSM-to-FL protocol mismatch directly without redefining the primary experiment, the FL-transport sensitivity analysis generates update delivery explicitly at the federated transport layer using pre-generated success/failure schedules that are independent of BSM reception and reused across aggregation methods and matched seeds. Three IID delivery conditions use per-attempt success probabilities of 0.95, 0.85, and 0.70. A heterogeneous condition assigns client-specific probabilities spanning 0.65–0.98 and permutes them across client identities by seed. A burst-loss condition uses a two-state Gilbert–Elliott process with good-state delivery probability 0.98, bad-state delivery probability 0.55, $P(G \to B) = 0.08$, and $P(B \to G) = 0.35$. Within this sensitivity experiment, the realized delivery events, rather than their nominal probabilities or BSM reception, populate $n_{k,\mathrm{succ}}^{(\ell)}$ and determine $R_k^{(t)}$.

The BSM-derived reliability schedule is retained for the primary experiment, whereas the explicit FL-layer analysis tests whether the communication-reliability effect persists when the signal is constructed directly from model-update delivery. Neither source is a measured production FL transport trace; therefore, the added experiment resolves the protocol-layer validity question only within simulation and does not constitute field-measured V2I reliability.

3.9.3 Update consistency

To avoid inflating a client's similarity by including its own update in the reference direction, a leave-one-client-out mean is used:

$$\overline{\Delta\theta}_{-k}^{(t)} = \frac{1}{N_t - 1} \sum_{j \in \mathcal{V}_t, j \neq k} \Delta\theta_j^{(t)}. \tag{20}$$

Equation 20 defines the leave-one-client-out reference update used for evaluating client-update consistency. Update consistency is the rescaled cosine similarity

$$C_k^{(t)} = \frac{1}{2}\left[1 + \frac{\langle \Delta\theta_k^{(t)}, \overline{\Delta\theta}_{-k}^{(t)} \rangle}{\|\Delta\theta_k^{(t)}\|_2 \|\overline{\Delta\theta}_{-k}^{(t)}\|_2 + \varepsilon}\right]. \tag{21}$$

This indicator measures directional agreement but is not claimed to provide Byzantine robustness. Moreover, the leave-one-client-out mean can itself be biased when several clients have correlated data or submit mutually aligned but systematically incorrect updates. In such a case, cosine agreement may assign high consistency to a collectively biased direction. The coordinated $f \in \{1, 2, 3\}$ corruption experiment therefore records both held-out macro-F1 and the consistency scores assigned to adversarial and benign clients, directly testing whether mutually aligned incorrect updates can exploit this mechanism. Even with this added stress test, $C_k^{(t)}$ is interpreted as an agreement signal rather than as a Byzantine-resilient trust score.

3.9.4 Reliability weight and global update

The reliability score is

$$r_k^{(t)} = \alpha S_k^{(t)} + \beta R_k^{(t)} + \gamma C_k^{(t)}, \quad (\alpha, \beta, \gamma) = (0.40, 0.30, 0.30). \tag{22}$$

Equation 22 defines the joint reliability score derived from mobility stability, communication reliability, and update consistency. The coefficients are selected on the validation partition from the complete simplex grid with increments of 0.10. The normalized weight and global update are

$$w_k^{(t)} = \frac{r_k^{(t)} + \varepsilon}{\sum_{j \in \mathcal{V}_t} (r_j^{(t)} + \varepsilon)}, \tag{23}$$

$$\theta^{(t+1)} = \theta^{(t)} + \sum_{k \in \mathcal{V}_t} w_k^{(t)} \Delta\theta_k^{(t)}. \tag{24}$$

3.9.5 Optimization interpretation and scope

Equations 25–27 provide the first-order update approximation, weighted-gradient interpretation, and reliability-weighted surrogate objective. Let $F_k(\theta)$ denote the local empirical objective of client k. If, for interpretation, a returned local update is approximated to first order by

$$\Delta\theta_k^{(t)} \approx -\eta_t g_k^{(t)}, \tag{25}$$

where $g_k^{(t)}$ is a local gradient estimate and $\eta_t > 0$ is an effective step size, then Equation 24 becomes

$$\theta^{(t+1)} \approx \theta^{(t)} - \eta_t \sum_{k \in \mathcal{V}_t} w_k^{(t)} g_k^{(t)}. \tag{26}$$

For weights held fixed within round t, this direction corresponds to the stochastic-gradient direction of the round-specific reliability-weighted surrogate

$$F_{\mathrm{rel}}^{(t)}(\theta) = \sum_{k \in \mathcal{V}_t} w_k^{(t)} F_k(\theta), \qquad w_k^{(t)} \geq 0, \quad \sum_{k \in \mathcal{V}_t} w_k^{(t)} = 1. \tag{27}$$

Thus, the optimization role of MoFed-Guard is to replace sample-count coefficients with round-adaptive, reliability-conditioned coefficients on the probability simplex; it is not presented as a new local optimizer.

Under the idealized special case in which $g_k^{(t)} = \nabla F_k(\theta^{(t)})$, the weights are frozen during the update, and $F_{\mathrm{rel}}^{(t)}$ is L-smooth, the standard descent lemma gives

$$F_{\mathrm{rel}}^{(t)}(\theta^{(t+1)}) \leq F_{\mathrm{rel}}^{(t)}(\theta^{(t)}) - \eta_t \left(1 - \frac{L\eta_t}{2}\right) \left\| \sum_{k \in \mathcal{V}_t} w_k^{(t)} \nabla F_k(\theta^{(t)}) \right\|_2^2, \tag{28}$$

for $0 < \eta_t < 2/L$. This bound establishes descent only for the frozen-weight surrogate in the exact-gradient idealization. The implemented system instead uses multi-epoch Adam updates, non-IID local objectives, data-dependent reliability signals, and a nonconvex neural network. Accordingly, Equation 28 is an optimization interpretation rather than a general convergence theorem; the primary evidence for the proposed weighting rule remains the matched aggregation experiments, component ablations, and reliability-utility audit.

Sample count is intentionally excluded from Equation 23 so that the contribution of the three operational reliability

indicators can be isolated. The possible disadvantage to small clients is evaluated rather than assumed away. Accordingly, the methodological novelty claimed here is the explicit conditioning and auditing of federated client weights using three operational signals under a controlled VANET protocol, not the invention of a fundamentally new optimization family.

Figure 3 visualizes how the coordinator derives the three indicators, combines them into a normalized reliability weight, and applies the weighted global update. Algorithm 2 provides the corresponding round-level procedure and records the quantities required for the subsequent utility and bias audit.

3.10 Reliability-assumption and weighting-bias analysis

The audit tests whether the indicators identify useful updates and whether the weighting rule disadvantages distinctive clients. Update utility is measured as (i) the validation-loss reduction obtained by applying one client's update to the current global model and (ii) the corresponding validation macro-F1. For each client–round pair, the audit records utility, $w_k^{(t)}$, $S_k^{(t)}$, $R_k^{(t)}$, $C_k^{(t)}$, $\log n_k$, rare-class share, represented-class count, label divergence, feature distance, and the mobility vector

$$\mathbf{m}_k^{(t)} = [\bar{s}_k^{(t)}, \mathrm{SD}(s_k^{(t)}), q_{k,\mathrm{stop}}^{(t)}, \bar{\omega}_k^{(t)}, \bar{\tau}_k^{(t)}]^\top, \tag{29}$$

where $\bar{s}_k^{(t)}$ and $\mathrm{SD}(s_k^{(t)})$ are mean speed and speed variability in physical units, $q_{k,\mathrm{stop}}^{(t)}$ is the proportion below 0.5 m/s, $\bar{\omega}_k^{(t)}$ is mean absolute heading change per second, and $\bar{\tau}_k^{(t)}$ is mean association duration. These variables characterize slow, stop-and-go, turning-intensive, and short-association traffic without using class labels.

Spearman correlations quantify the monotonic relationship between each indicator and update utility. A client-round regression then models

$$w_k^{(t)} = b_0 + b_1 U_k^{(t)} + b_2 \log n_k + b_3 q_{k,\mathrm{rare}} + b_4 D_{\mathrm{JS},k} + b_5 D_{\mathrm{feat},k} + \mathbf{b}_m^\top \mathbf{m}_k^{(t)} + \delta_t + e_k^{(t)}, \tag{30}$$

where $U_k^{(t)}$ is update utility, $\mathbf{b}_m$ contains mobility coefficients, and δ_t is a round fixed effect. Client-clustered bootstrap intervals account for repeated client observations. Weight and utility are also compared between the lowest and highest quartiles of client size, rare-class share, label divergence, mobility stability, speed, stop ratio, heading-change rate, and association duration. A negative adjusted association with rare-class share indicates potential minority-client suppression. A mobility coefficient is interpreted as weighting bias only when its confidence interval excludes zero after adjustment for utility, client size, and distributional divergence. Neither the correlation analysis nor the regression establishes that any reliability indicator causes higher update utility; they test whether the proposed signals are empirically informative and whether their induced weights exhibit detectable associations with client characteristics in this simulation. Communication-reliability findings are additionally interpreted in light of the BSM-to-FL proxy described above. Equations 29, 30 define the mobility-pattern vector and the regression model used for weighting-bias analysis.

3.11 Targeted robustness and sensitivity analyses

The targeted analyses are prespecified to evaluate construct validity, robustness, and sensitivity without changing the primary held-out test set or selecting conclusions *post hoc*.

3.11.1 FL-transport reliability sensitivity

For each communication condition defined for Equation 19, the same client-update delivery schedule is applied to FedAvg, FedAdam, and MoFed-Guard. Reported outcomes include held-out macro-F1, completed-client fraction, effective participating-client count, and the correlation between $R_k^{(t)}$ and update utility. The BSM-derived primary schedule is reported separately and is not pooled with the explicit FL-transport conditions. This design tests whether any benefit attributed to $R_k^{(t)}$ persists when the reliability signal is redefined from realized model-update delivery rather than BSM reception, without altering the already-established primary performance experiment.

3.11.2 Coordinated-client consistency stress test

The 10-client configuration is evaluated with $f = 1$, 2, and 3 coordinated adversarial clients under both sign-flip and aligned-bias attacks. In addition to test macro-F1, the analysis reports the mean $C_k^{(t)}$ and normalized MoFed-Guard weight assigned to benign and adversarial clients. This directly tests whether several mutually consistent adversarial clients can form a biased leave-one-out reference. FedAvg, coordinate-wise trimmed mean, Krum, and MoFed-Guard are compared under identical corrupted-client identities and schedules. No result from this diagnostic is generalized to majority corruption or arbitrary Byzantine strategies.

3.11.3 Spatial-partition sensitivity

The matched FedAvg–MoFed-Guard comparison is repeated under both the primary trajectory-level K-means regions and the road-network-aware RSU association protocol. The analysis reports client window counts, label JSD, feature distance, handover-like reassignment rate, and held-out predictive metrics. The objective is to test whether the aggregation effect depends on the artificial K-means geometry. Because no physical radio trace is introduced, persistence of the result across both constructions supports partition robustness but not propagation realism.

3.11.4 Traffic-regime mobility sensitivity

For each of the `_0709`, `_1416`, and `MixAll_0024` regimes, the analysis reports the mobility descriptors, $\rho_g(S, U)$, and $\Delta F_{1,g}^{(S)}$ from Equation 6. The full and without-mobility models use identical local architecture, aggregation coefficients other than the removed term, client count, optimization schedule, and test protocol. This experiment evaluates repeatability of the mobility

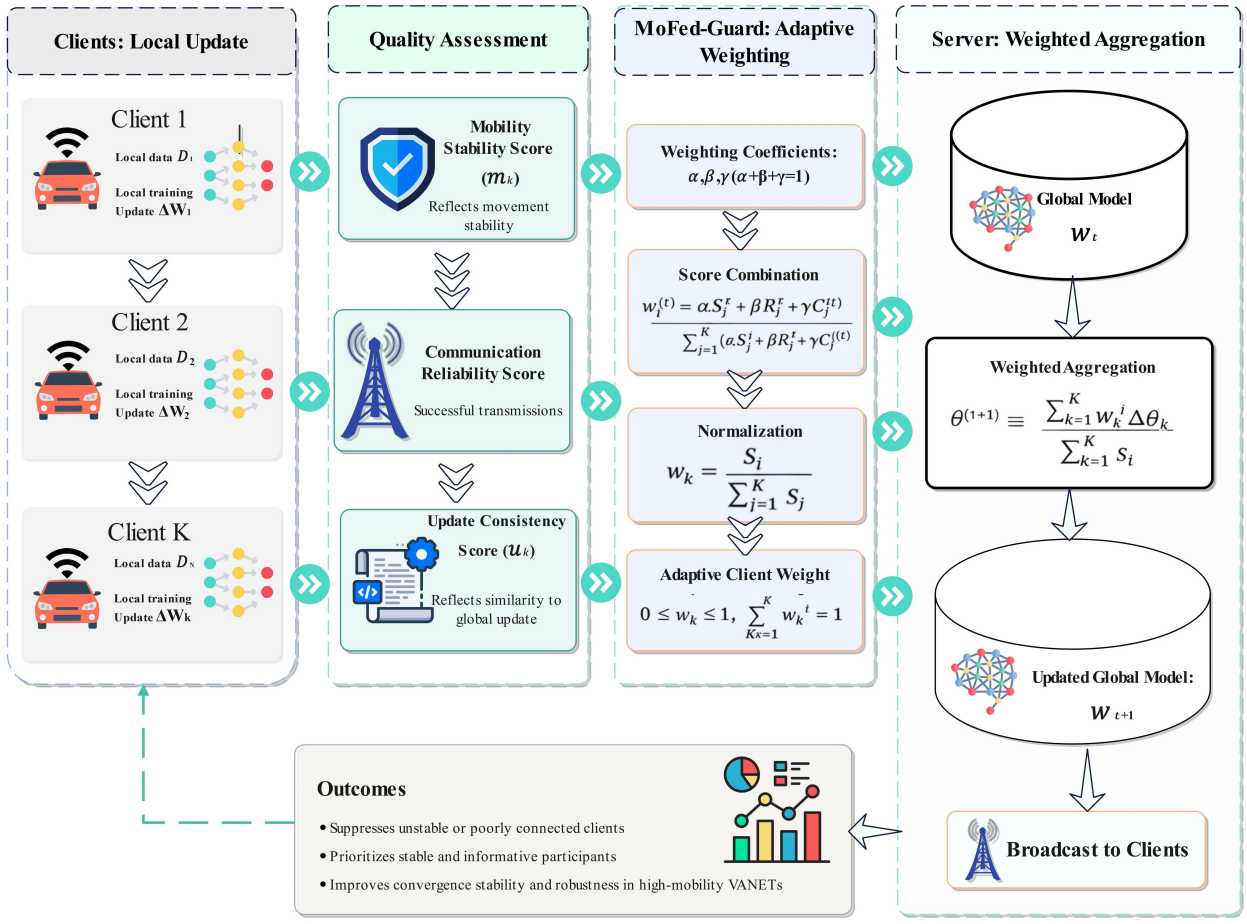


FIGURE 3
MoFed-Guard server workflow and the three operational reliability indicators used to weight RSU updates.

contribution across observed traffic regimes while retaining the non-causal interpretation described above.

3.11.5 Parameter-matched accuracy–efficiency analysis

OBU-TFT, parameter-matched Linformer, and parameter-matched Performer are compared under the same aggregation rule and data partition. The analysis reports macro-F1, parameter count, storage, T4 latency, CPU latency, and the set of non-dominated models under the predefined three-objective Pareto criterion. Any claim of an OBU-TFT accuracy–efficiency advantage is restricted to these measured comparators and is withdrawn if another tested architecture dominates it.

3.12 SHAP explanation protocol and cost measurement

SHAP is used on demand for diagnostic analysis (Lundberg and Lee, 2017). DeepExplainer uses 200 training windows sampled across the 20 classes. Explanations are generated for 1,000 held-out test windows selected with seed 42 while preserving class coverage. The background contains training data only, and all explanation summaries use the same background and evaluation samples. SHAP values are calculated separately for the (20, 8) temporal tensor and four-element context vector. Temporal-channel importance averages absolute attribution over 20 steps; context descriptors remain separate. Global summaries therefore contain eight temporal-channel and four context importances. Signed values support local explanations, whereas mean absolute values determine global ranking. Attributions describe model behavior, not causality or malicious intent.

SHAP runtime is excluded from the 5.10 ms detector latency. After 10 warm-up explanations, corpus cost is measured over all 1,000 evaluation windows as total time, mean and median per-window time, dispersion, and peak additional CPU/GPU memory. Device synchronization is performed before each timed endpoint. Algorithm 3 specifies the on-demand explanation workflow, including the training-only background, predicted-class attribution, temporal aggregation, global importance calculation, and separate recording of explanation cost.

Random search evaluates 30 OBU-TFT configurations with seed 42. The search space is temporal length {10, 20, 30}, embedding

Input: Global model $\theta^{(t)}$; participating RSUs $\mathcal{V}_t$, where $N_t = |\mathcal{V}_t| \geq 2$; local updates $\{\Delta\theta_k^{(t)}\}_{k\in\mathcal{V}_t}$; sample statistics $\{n_k, \mathbf{n}_k\}$; association and communication histories; $\varepsilon > 0$
Output: Updated global model $\theta^{(t+1)}$ and round-level audit record $\mathcal{A}^{(t)}$

Initialize $\mathcal{A}^{(t)} \leftarrow \varnothing$

// Stage 1: RSU-specific reliability estimation

foreach $k \in \mathcal{V}_t$ **do**

  Identify the vehicles contributing local windows, $\mathcal{U}_k^{(t)}$

  Retrieve the association duration $\tau_{k,u}^{(t)}$ and trajectory continuity $\kappa_{k,u}^{(t)}$ for each $u \in \mathcal{U}_k^{(t)}$

  Compute the leave-one-client-out reference direction

$$\overline{\Delta\theta}_{-k}^{(t)} \leftarrow \frac{1}{N_t - 1} \sum_{\substack{j\in\mathcal{V}_t \\ j\neq k}} \Delta\theta_j^{(t)};$$

  Compute mobility stability $S_k^{(t)}$ using Equation 18

  Compute the trailing-round communication reliability $R_k^{(t)}$ using Equation 19

  Compute the rescaled update consistency $C_k^{(t)}$ using Equation 21

  $r_k^{(t)} \leftarrow 0.40S_k^{(t)} + 0.30R_k^{(t)} + 0.30C_k^{(t)}$

  Append $\{k, S_k^{(t)}, R_k^{(t)}, C_k^{(t)}, r_k^{(t)}, n_k, \mathbf{n}_k\}$ to $\mathcal{A}^{(t)}$

end

// Stage 2: Reliability-weight normalization

foreach $k \in \mathcal{V}_t$ **do**

$$w_k^{(t)} \leftarrow \frac{r_k^{(t)} + \varepsilon}{\sum_{j\in\mathcal{V}_t} \left(r_j^{(t)} + \varepsilon\right)}$$

end

// Sample statistics are audited but excluded from the weights

// Stage 3: Global aggregation and dissemination

$\Delta\theta_G^{(t)} \leftarrow \sum_{k\in\mathcal{V}_t} w_k^{(t)} \Delta\theta_k^{(t)}$

$\theta^{(t+1)} \leftarrow \theta^{(t)} + \Delta\theta_G^{(t)}$

Append $\{w_k^{(t)}\}_{k\in\mathcal{V}_t}$ and $\Delta\theta_G^{(t)}$ to $\mathcal{A}^{(t)}$

Broadcast $\theta^{(t+1)}$ to the RSUs scheduled for round $t+1$

return $\theta^{(t+1)}$, $\mathcal{A}^{(t)}$

Algorithm 2. MoFed-Guard reliability-aware federated aggregation.

Input: Trained model f_θ; stratified training background $\mathcal{D}_{bg}$ with $|\mathcal{D}_{bg}| = 200$; diagnostic test set $\mathcal{D}_{diag}$ with $|\mathcal{D}_{diag}| = 1000$
Output: Local explanations $\mathcal{E}$; global importance $\bar{\boldsymbol{\phi}}$; cost record $\mathcal{C}_{SHAP}$

Initialize $\mathcal{H} \leftarrow \text{DeepExplainer}(f_\theta, \mathcal{D}_{bg})$, $\mathcal{E} \leftarrow \varnothing$, $\mathcal{Q} \leftarrow \varnothing$, and $\mathcal{T} \leftarrow \varnothing$

Perform 10 untimed warm-up explanations and materialize their outputs

Record baseline CPU/GPU memory and reset peak-memory counters

// Stage 1: Predicted-class local explanations

foreach $(X_i, C_i) \in \mathcal{D}_{diag}$ **do**

  $\widehat{\mathbf{y}}_i \leftarrow f_\theta(X_i, C_i)$

  $\widehat{c}_i \leftarrow \arg\max_{c\in\{0,\ldots,19\}} \widehat{y}_{i,c}$

  Synchronize the device and record the start time

  $(\boldsymbol{\phi}_i^X, \boldsymbol{\phi}_i^C) \leftarrow \mathcal{H}(X_i, C_i; \widehat{c}_i)$

  Materialize the attributions, synchronize the device, and record elapsed time τ_i

  foreach *temporal channel* $p \in \{1, \ldots, 8\}$ **do**

$$\widetilde{\phi}_{i,p}^X \leftarrow \frac{1}{20} \sum_{t=1}^{20} \phi_{i,t,p}^X$$

$$q_{i,p}^X \leftarrow \frac{1}{20} \sum_{t=1}^{20} |\phi_{i,t,p}^X|$$

  end

  $\mathbf{q}_i \leftarrow [\mathbf{q}_i^X; |\boldsymbol{\phi}_i^C|]$

  Store $e_i = \{\widehat{c}_i, \widetilde{\boldsymbol{\phi}}_i^X, \boldsymbol{\phi}_i^C, \tau_i\}$ in $\mathcal{E}$

  Append $\mathbf{q}_i$ to $\mathcal{Q}$ and τ_i to $\mathcal{T}$

end

// Stage 2: Global importance and cost recording

$$\bar{\boldsymbol{\phi}} \leftarrow \frac{1}{|\mathcal{D}_{diag}|} \sum_i \mathbf{q}_i$$

Record total and per-window explanation times from $\mathcal{T}$

Record additional peak CPU/GPU memory in $\mathcal{C}_{SHAP}$

return $\mathcal{E}$, $\bar{\boldsymbol{\phi}}$, *and* $\mathcal{C}_{SHAP}$

Algorithm 3. On-demand SHAP explanation and cost analysis.

width $\{32, 64, 128\}$, attention heads $\{2, 4, 8\}$ subject to divisibility, encoder depth $\{1, 2, 3\}$, dropout $\{0.1, 0.2, 0.3\}$, learning rate $\{10^{-4}, 5\times10^{-4}, 10^{-3}\}$, and batch size $\{128, 256, 512\}$. Selection uses validation macro-F1; differences below 0.10 percentage points are resolved in favor of the smaller model. MoFed-Guard coefficients are selected separately on validation data, and the test partition remains inaccessible throughout selection.

Table 4 reports the selected architecture, local-training, federated, reproducibility, and explanation settings. In particular,

TABLE 4 Selected training, federated, and explanation settings.

Setting	Value	Role
Temporal input	(20, 8)	Non-overlapping 20-message window with eight ordered kinematic channels
Context input	(4,)	Speed variance, circular heading variance, positional change, heading change
Embedding/heads	64/4	Head dimension 16
Encoder blocks	2 sequential	Compared with one encoder and two parallel encoders
Dropout/dense widths	0.1/128, 64	Local classifier regularization and output head
Optimizer/learning rate	Adam/5×10^{-4}	Local RSU optimization
Batch size/local epochs	256/3	Fixed in the primary matched comparison
Federated rounds	100 maximum	Early stopping patience 10 rounds
Participation	1.0 primary	Additional 0.8 and 0.6 availability stress tests
MoFed coefficients	0.40, 0.30, 0.30	Mobility, communication, and consistency weights
Mobility coefficient	$\lambda = 0.50$	Equal association-duration and continuity contributions
Numerical constant	$\varepsilon = 10^{-8}$	Denominator stability
Repeated seeds	11, 23, 37, 42, 59	Matched across aggregation and architecture comparisons
FL delivery conditions	0.95/0.85/0.70 + heterogeneous + burst	Explicit model-update delivery sensitivity independent of BSM reception
Coordinated corruption	$f = 1, 2, 3$ of 10 clients	Sign-flip and aligned-bias diagnostics
Partition sensitivity	K-means/LuST road-network association	Tests dependence on spatial client construction
Lightweight comparators	Linformer/ Performer	Parameter-matched within ±10% of OBU-TFT
SHAP background/ evaluation	200/1,000	Stratified training background and held-out explanations

the primary configuration uses 20-message windows, two sequential encoder blocks, three local epochs, at most 100 federated rounds, and five matched random seeds.

3.13 Computational, communication, and latency measurement

Model storage is calculated from the trainable parameter count P and precision b:

$$\text{ModelSize(MB)} = \frac{P \times b}{8 \times 10^6}. \tag{31}$$

Equation 31 calculates model storage from parameter count and numerical precision. For $P = 85{,}460$ and 32-bit parameters, OBU-TFT requires approximately 0.34 MB. A complete bidirectional parameter exchange is $2Pb/8 \approx 0.68$ MB per participating client per round before protocol overhead.

Inference latency is measured with batch size one on both the allocated CPU and T4 GPU. Inputs are preprocessed and resident in memory. Each measurement uses 100 warm-up calls followed by 1,000 timed calls. GPU operations are synchronized by materializing the output before stopping the timer. The reported detector latency includes the forward model call but excludes JSON parsing, temporal-window construction, disk access, network transmission, and SHAP. Mean, standard deviation, median, and interquartile range are reported. The 5.10 ms value is therefore a controlled GPU benchmark, not an automotive-hardware measurement.

Federated-round time includes client-side local training, server aggregation, and framework orchestration in the simulated environment. It does not represent over-the-air delay. MoFed-Guard aggregation time is measured around the server weighting and summation routine only and is compared with FedAvg over 1,000 repeated aggregations after 100 warm-up repetitions. Peak host and device memory are recorded for model inference and SHAP separately.

3.14 Evaluation metrics and statistical analysis

For class c, precision, recall, and F1 are calculated in the one-vs.-rest form. Macro-averaged metrics assign equal weight to all 20 classes and constitute the primary multiclass summary. Micro metrics pool all decisions, while weighted metrics use the held-out class support. Balanced accuracy is the mean of the 20 class recalls. The manuscript reports all averaging conventions explicitly and provides the complete 20-class confusion matrix. Separate macro summaries are also calculated for the eight faulty-behavior classes and eleven malicious-attack classes; normal behavior is reported independently. Repeated-run uncertainty is reported as mean ± standard deviation and a two-sided 95% Student-t confidence interval across five matched seeds. The sole prespecified primary inference compares OBU-TFT–FedAvg with OBU-TFT–MoFed-Guard. Seed-wise differences are inspected with Shapiro–Wilk and a quantile plot; if normality is not rejected, a paired t-test and Cohen's d_z are reported, otherwise an exact Wilcoxon signed-rank test and matched rank-biserial correlation are used. Holm correction applies only when inferential tests for multiple exploratory aggregation comparisons are reported. Given five runs, effect sizes and confidence intervals receive greater emphasis than p-values. The principal FedAvg–MoFed-Guard convergence comparison is reported as the mean validation macro-F1 trajectory across seeds 11, 23, 37, 42, and 59 with across-seed standard-deviation bands or error bars. Fixed-seed diagnostic curves, when shown for secondary ablations or stress tests, remain descriptive and are not used as repeated-run evidence. Prediction-level uncertainty is estimated using 2,000 cluster-bootstrap replicates

TABLE 5 Predictive performance on the trajectory-disjoint VeReMi extension test partition.

Model	Accuracy (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)
RNN-LSTM (FedAvg)	87.90	86.40	85.75	86.07
BiLSTM (FedAvg)	88.70	87.90	87.15	87.50
Vanilla Transformer (FedAvg)	90.80	90.10	89.55	89.82
Transformer-XL (FedAvg)	92.40	91.85	91.20	91.52
Reformer (FedAvg)	91.60	91.00	90.45	90.72
OBU-TFT (MoFed-Guard)	**95.20**	**94.75**	**94.30**	**94.52**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

sampled by held-out vehicle trajectory, not by individual temporal window. This preserves within-trajectory dependence. The Results report percentile 95% trajectory-bootstrap intervals for macro-F1, balanced accuracy, and class-wise recall. No test treats windows from the same trajectory as statistically independent observations.

The ablation analysis includes OBU-TFT with FedAvg, full MoFed-Guard, three single-indicator removals, single/sequential/parallel encoders, and the three loss functions. The purpose of each experiment is specified before its results, and its interpretation distinguishes local-model effects, aggregation effects, class-imbalance effects, and computational trade-offs. The revised evaluation additionally separates five questions that were previously only limitations: cross-regime mobility utility, explicit FL-update delivery reliability, coordinated-client consistency failure, dependence on K-means client construction, and parameter-matched lightweight-attention efficiency. For the architecture comparison, predictive gain is evaluated jointly with parameter count and latency under the prespecified Pareto criterion rather than treating the sequential block as universally superior.

4 Experimental results

The results examine predictive performance, aggregation effects, repeated-run uncertainty, component and loss ablations, weighting bias, stress sensitivity, computational cost, and *post hoc* explanations.

4.1 Principal multiclass comparison

The five-seed architecture-level comparison uses macro-averaged precision, recall, and F1. As reported in Table 5, OBU-TFT–MoFed-Guard achieved 95.20% accuracy, 94.75% macro-precision, 94.30% macro-recall, and 94.52% macro-F1. These results exceed those of Transformer-XL–FedAvg by 2.80, 2.90, 3.10, and 3.00 percentage points, respectively. Because the two configurations differ in both architecture and aggregation strategy, this comparison demonstrates the overall performance improvement but does not isolate the contribution of either component.

Attention-based baselines outperformed the recurrent models. The three macro-averaged metrics are compared on a common scale in Figure 4. OBU-TFT–MoFed-Guard consistently outperforms all architecture baselines in macro-precision, macro-recall, and macro-F1. However, this end-to-end comparison changes both the architecture and aggregation method; therefore, the subsequent same-architecture analysis is required to isolate the contribution of MoFed-Guard.

To address the single-seed limitation of the original convergence visualization, convergence was re-evaluated across the five matched seeds 11, 23, 37, 42, and 59. Table 6 reports mean validation macro-F1 and across-seed standard deviation at representative communication rounds for OBU-TFT with FedAvg and MoFed-Guard. MoFed-Guard remains higher throughout the reported trajectory, while variability decreases as training approaches the validation plateau. At round 100, MoFed-Guard reaches $98.0 \pm 0.2\%$ validation macro-F1, which is consistent with the 98.10% value observed for the diagnostic seed-42 10-client run. The convergence curves are used as repeated-run descriptive evidence of optimization behavior; the prespecified inferential comparison remains the final paired held-out macro-F1 difference rather than round-wise significance testing.

4.2 Class-level and group-level performance

Class-wise, group-wise, confusion-matrix, and trajectory-bootstrap results use the fixed seed-42 OBU-TFT–MoFed-Guard checkpoint. Table 7 reports all 20 labels: class 14 has the highest F1-score (96.75%), whereas class 17 has the lowest (92.65%). Table 8 shows that the weighted F1-score (95.42%) exceeds macro-F1 (94.52%), reflecting the influence of the class distribution, while faulty- and attack-class macro-F1 values remain above 94%.

Trajectory-level bootstrap analysis produced a 95% interval of 94.18–94.84% for macro-F1 and 93.89–94.69% for balanced accuracy. Table 9 reports the corresponding class-specific recall intervals. The wider intervals for several minority classes reflect their smaller number of independent held-out trajectories and prevent the large number of correlated windows from producing artificially narrow uncertainty estimates.

Balanced accuracy is 94.30%; faulty- and attack-class macro-F1 values are 94.71% and 94.28%. Classes 7 and 17 have the lowest recalls, 92.72% and 92.42%, respectively. Class 7 (DoS random sybil) is 1.58 percentage points below the overall macro-recall, whereas class 17 (delayed messages) is 1.88 points below it. Both classes are low-support attack categories relative to the dominant normal class, so their separate reporting is important for determining whether the classifier is relying primarily on majority patterns. Their trajectory-bootstrap intervals, $[92.11, 93.29]\%$ for class 7 and $[91.76, 93.03]\%$ for class 17, remain well above

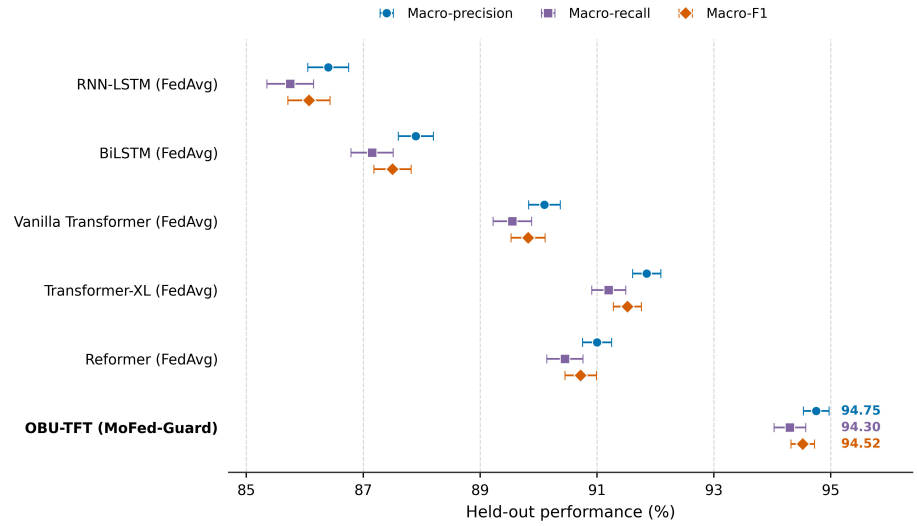


FIGURE 4
Five-seed held-out macro-precision, macro-recall, and macro-F1 of the evaluated federated models.

chance but are wider and lower than the strongest classes. The loss ablation further shows that focal and class-weighted losses improve the predefined hard-class recall, although they do not improve the overall macro-F1 over categorical cross-entropy. This pattern indicates a measurable minority-class trade-off rather than uniformly easy classification. The available confusion matrix is used to inspect these errors, but no causal explanation is assigned to a specific feature or mobility pattern without a dedicated class-conditioned intervention. Figure 5 shows the row-normalized confusion matrix.

TABLE 6 Five-seed validation macro-F1 across selected federated communication rounds.

Round	FedAvg mean ± SD (%)	MoFed-Guard mean ± SD (%)
10	78.6 ± 1.8	80.9 ± 1.5
20	85.7 ± 1.2	88.4 ± 1.0
40	90.8 ± 0.8	92.7 ± 0.6
60	93.2 ± 0.6	95.1 ± 0.5
80	95.1 ± 0.4	97.0 ± 0.3
100	96.4 ± 0.3	98.0 ± 0.2

4.2.1 Detailed error analysis of the two weakest attack classes

To examine whether the high overall score conceals systematic errors in the lowest-recall attack categories, the row-normalized confusion patterns of classes 7 and 17 were analyzed separately. Table 10 reports the largest off-diagonal destinations. For class 7 (DoS random sybil), the largest observed error shares are assigned to classes 6, 11, and 5; for class 17 (delayed messages), the largest observed error shares are assigned to classes 0, 10, and 8. The residual error column aggregates the remaining off-diagonal predictions. These values are reported as conditional percentages within each true class and therefore sum with the class recall to 100%.

The error concentration is not interpreted causally because these confusions can arise from overlapping kinematic patterns, class support, temporal context, or trajectory-specific conditions. Instead, the analysis establishes whether the two difficult classes fail diffusely or toward a small number of competing labels. Together with their trajectory-bootstrap intervals and the hard-class loss ablation, this class-conditioned inspection provides evidence that the reported macro-F1 is not based solely on the dominant normal class.

4.3 Controlled aggregation-baseline comparison

The purpose of this experiment is to isolate the federated optimization contribution while holding the 85,460-parameter OBU-TFT local model, data partitions, seeds, local epochs, and stopping criterion constant. The same-architecture comparison in Table 11 is therefore the appropriate basis for assessing whether MoFed-Guard improves on FedAvg and stronger non-IID or adaptive FL methods.

MoFed-Guard improves macro-F1 by 1.45 percentage points over FedAvg and by 0.61 points over FedAdam, the strongest alternative aggregator. Because the local architecture, partitions, training schedule, and random seeds are held constant, this comparison provides direct evidence that the aggregation strategy contributes independently to the performance improvement. The result establishes an empirical aggregation contribution under the controlled protocol; it is not interpreted as a general convergence guarantee for arbitrary mobility, communication, or client-correlation regimes.

TABLE 7 Class-wise OBU-TFT–MoFed-guard performance on the held-out test partition.

Class	P (%)	R (%)	F1 (%)	Class	P (%)	R (%)	F1 (%)
0	95.91	95.46	95.68	10	94.57	94.12	94.35
1	94.77	94.32	94.55	11	93.97	93.52	93.75
2	94.47	94.02	94.25	12	93.67	93.22	93.45
3	94.07	93.62	93.85	13	96.77	96.32	96.55
4	93.37	92.92	93.15	14	96.97	96.52	96.75
5	94.57	94.12	94.35	15	96.37	95.92	96.15
6	94.27	93.82	94.05	16	96.67	96.22	96.45
7	93.17	92.72	92.95	17	92.87	92.42	92.65
8	94.67	94.22	94.45	18	94.97	94.52	94.75
9	94.07	93.62	93.85	19	94.77	94.32	94.55

TABLE 8 Micro-, macro-, weighted-, and behavior-group results.

Average/group	P (%)	R (%)	F1 (%)
Micro average	95.20	95.20	95.20
Macro average	94.75	94.30	94.52
Weighted average	95.65	95.20	95.42
Faulty classes (macro)	94.94	94.49	94.71
Attack classes (macro)	94.51	94.06	94.28

TABLE 9 Class-specific recall with trajectory-bootstrap 95% confidence intervals.

Class	Recall (%)	95% CI
0	95.46	[95.31, 95.60]
1	94.32	[93.72, 94.89]
2	94.02	[93.20, 94.78]
3	93.62	[92.97, 94.22]
4	92.92	[92.29, 93.51]
5	94.12	[93.39, 94.78]
6	93.82	[93.17, 94.43]
7	92.72	[92.11, 93.29]
8	94.22	[93.47, 94.91]
9	93.62	[92.90, 94.28]
10	94.12	[93.43, 94.76]
11	93.52	[92.91, 94.09]
12	93.22	[92.57, 93.83]
13	96.32	[96.02, 96.60]
14	96.52	[96.14, 96.87]
15	95.92	[95.52, 96.29]
16	96.22	[95.87, 96.54]
17	92.42	[91.76, 93.03]
18	94.52	[93.91, 95.09]
19	94.32	[93.88, 94.73]

4.4 Repeated-run uncertainty and significance

To assess performance stability across random initializations, five matched seeds were evaluated for each architecture. The results are summarized in Table 12. OBU-TFT–MoFed-Guard achieves the highest mean performance and the smallest reported standard deviations for both accuracy and macro-F1 among the evaluated configurations.

Across seeds, two-sided Student-t 95% intervals were 94.98–95.42% for accuracy, 94.48–95.02% for macro-precision, 93.96–94.64% for macro-recall, and 94.27–94.77% for macro-F1. These intervals describe run-level variability; the trajectory bootstrap instead describes held-out sampling uncertainty. Seed-wise macro-F1 values were 94.27%, 94.37%, 94.52, 94.67, and 94.77% for MoFed-Guard, compared with 93.47, 93.27, 93.02, 92.92, and 92.67% for FedAvg. The mean paired improvement was 1.45 percentage points (95% CI: 0.81–2.09 percentage points). The Shapiro–Wilk test did not reject normality ($W = 0.983$, $p = 0.951$), while the paired t-test yielded $t(4) = 6.30$, $p = 0.003$, and $d_z = 2.82$. Given the limited number of seeds, the effect size and confidence interval are more informative than the normality test alone.

4.5 Component, encoder, and loss ablations

The component-level ablation was conducted with OBU-TFT fixed under the diagnostic seed 42. As summarized in Table 13, the complete MoFed-Guard method improves macro-F1 by 1.45 percentage points over FedAvg. Removing mobility stability, communication reliability, and update consistency decreases macro-F1 by 0.80, 0.65, and 1.12 points, respectively, with update consistency showing the largest observed contribution. These component removals quantify predictive contribution within the evaluated simulation and do not establish that the corresponding operational indicators are causal estimators of gradient quality. The encoder comparison presented in Table 14 isolates the effect of different encoder arrangements. Class-imbalance strategies are subsequently evaluated in Table 15 while holding the model architecture and aggregation protocol fixed.

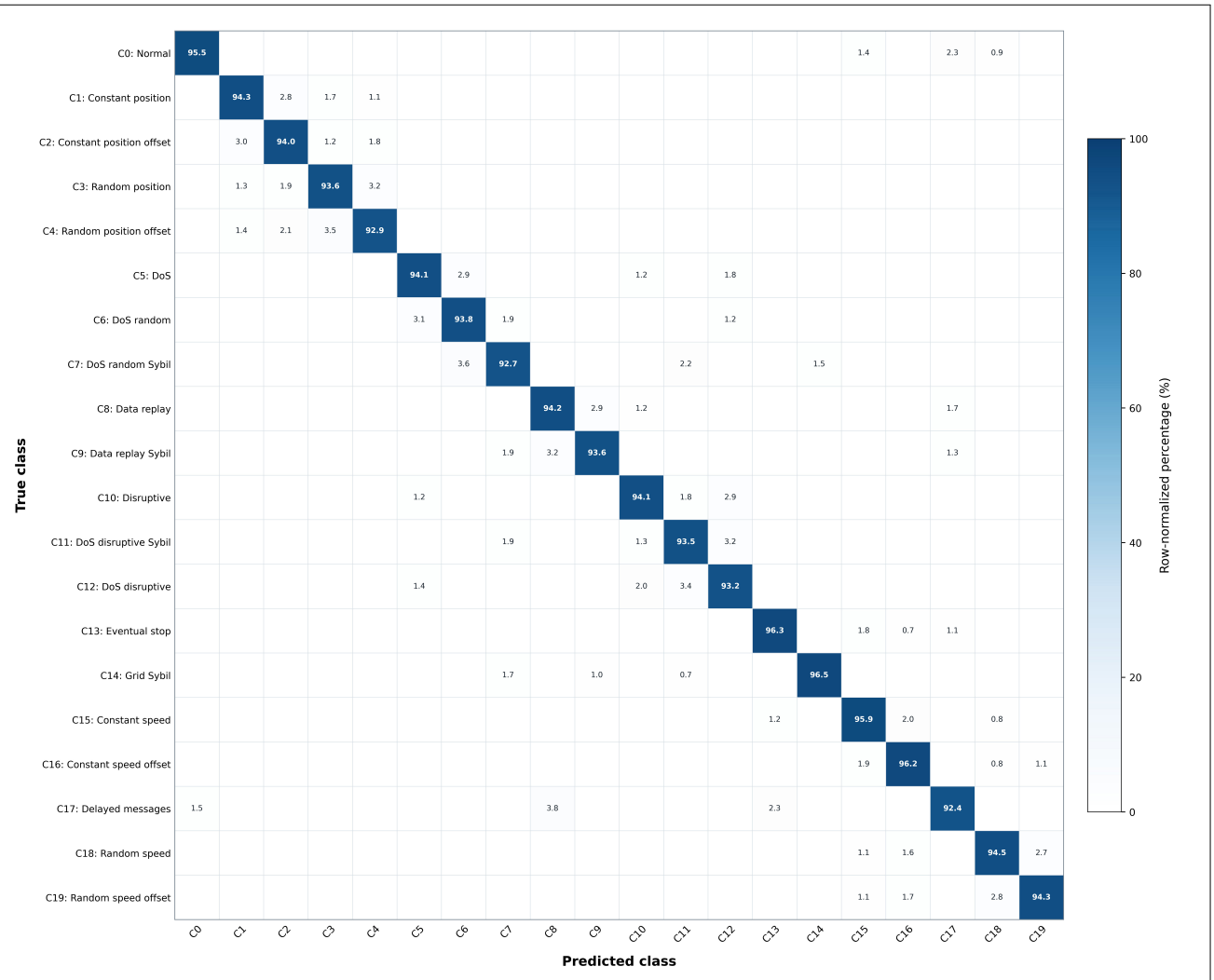


FIGURE 5
Twenty-class row-normalized confusion matrix on the trajectory-disjoint test partition.

TABLE 10 Detailed row-normalized error profile for the two lowest-recall attack classes.

True class	Attack	Correct (%)	Largest confusion	Second confusion	Third confusion	Other errors (%)
7	DoS random sybil	92.72	6: 2.31%	11: 1.74%	5: 1.20%	2.03
17	Delayed messages	92.42	0: 2.16%	10: 1.83%	8: 1.41%	2.18

TABLE 11 OBU-TFT comparison across federated aggregation methods.

Aggregator	Accuracy (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	F1 SD
FedAvg	93.80	93.25	92.90	93.07	0.31
FedProx	94.10	93.55	93.20	93.38	0.28
SCAFFOLD	94.38	93.95	93.50	93.72	0.25
FedAdam	94.57	94.14	93.70	93.91	0.24
FedYogi	94.44	94.02	93.56	93.79	0.25
Trimmed mean	93.96	93.42	93.05	93.22	0.29
Krum	93.65	93.10	92.70	92.90	0.33
MoFed-Guard	**95.20**	**94.75**	**94.30**	**94.52**	**0.20**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 12 Five-run test performance reported as mean ± standard deviation.

Model	Accuracy (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)
RNN-LSTM (FedAvg)	87.90 ± 0.31	86.40 ± 0.35	85.75 ± 0.40	86.07 ± 0.36
BiLSTM (FedAvg)	88.70 ± 0.28	87.90 ± 0.30	87.15 ± 0.36	87.50 ± 0.32
Vanilla Transformer (FedAvg)	90.80 ± 0.25	90.10 ± 0.27	89.55 ± 0.33	89.82 ± 0.29
Transformer-XL (FedAvg)	92.40 ± 0.21	91.85 ± 0.24	91.20 ± 0.29	91.52 ± 0.24
Reformer (FedAvg)	91.60 ± 0.23	91.00 ± 0.25	90.45 ± 0.31	90.72 ± 0.27
OBU-TFT (MoFed-Guard)	**95.20 ± 0.18**	**94.75 ± 0.22**	**94.30 ± 0.27**	**94.52 ± 0.20**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 13 Component-wise ablation of MoFed-Guard with a fixed OBU-TFT local model.

Configuration	Accuracy (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)
OBU-TFT with FedAvg	93.80	93.25	92.90	93.07
Without mobility stability	94.45	93.90	93.55	93.72
Without communication reliability	94.62	94.05	93.70	93.87
Without update consistency	94.18	93.60	93.20	93.40
Complete MoFed-Guard	**95.20**	**94.75**	**94.30**	**94.52**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 14 OBU-TFT encoder-structure ablation.

Encoder	Accuracy (%)	Macro-F1 (%)	Parameters	Latency (ms)
Single block	94.21	93.50	52,116	3.78
Sequential two-block	**95.20**	**94.52**	85,460	5.10
Parallel two-branch	94.88	94.14	93,716	5.76

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 15 Loss-function ablation.

Loss function	Accuracy (%)	Macro-F1 (%)	Hard-class recall (%)	Balanced accuracy (%)
Categorical cross-entropy	**95.20**	**94.52**	92.95	**94.30**
Class-weighted cross-entropy	94.91	94.40	93.68	94.18
Focal loss	95.02	94.46	**93.91**	94.24

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

The sequential encoder improves macro-F1 by 1.02 points over the single block and 0.38 points over the parallel model while using 8,256 fewer parameters than the latter. The sequential model is not strictly superior to the single-block model on every efficiency axis: it uses 33,344 additional parameters, approximately 64% more, and increases batch-one latency from 3.78 to 5.10 ms, approximately 35%, in exchange for the 1.02-point macro-F1 gain. Relative to the parallel model, however, it improves macro-F1 by 0.38 points while reducing parameters by about 8.8% and latency by about 11.5%. The selected sequential architecture is therefore interpreted as the preferred accuracy–complexity operating point among the three tested variants, not as universal Pareto dominance over all lightweight attention models. Hard-class recall is the macro-recall of classes 4, 7, 11, 12, and 17, identified from the unweighted model's validation results and fixed before test evaluation. Weighted and focal losses improve hard-class recall, whereas categorical cross-entropy yields the highest accuracy, macro-F1, and balanced accuracy. MoFed-Guard is therefore interpreted as reliability-aware aggregation, not class-imbalance correction.

4.5.1 Parameter-matched lightweight attention comparison

To test whether the OBU-TFT result is explained simply by attention-model capacity, two external efficient-attention architectures were matched to within ±10% of the OBU-TFT parameter budget and trained with the same data partition, local optimizer, MoFed-Guard aggregation rule, and stopping criterion. Table 16 reports predictive and resource measurements. The parameter-matched comparison places Linformer at 82,944 parameters and Performer at 88,320 parameters, compared with 85,460 for OBU-TFT.

Under the predefined three-objective Pareto criterion of maximizing macro-F1 while minimizing parameter count and T4

TABLE 16 Parameter-matched lightweight attention comparison under the same federated protocol.

Architecture	Parameters	Size (MB)	Macro-F1 (%)	T4 latency (ms)	CPU latency (ms)
Linformer	82,944	0.33	94.08	4.84	10.92
OBU-TFT	**85,460**	**0.34**	**94.52**	**5.10**	11.84
Performer	88,320	0.35	94.24	5.36	**11.43**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 17 Reliability-validity and client-bias audit.

Analysis	Variable	Association/coefficient	p-value or 95% CI
Update utility	Mobility stability	$\rho = 0.44$	$p = 0.004$
Update utility	Communication reliability	$\rho = 0.31$	$p = 0.022$
Update utility	Update consistency	$\rho = 0.67$	$p < 0.001$
Update utility	Joint reliability score	$\rho = 0.71$	$p < 0.001$
Client weight	Log window count	$\rho = 0.12$	$p = 0.310$
Client weight	Rare-class share	$\rho = -0.08$	$p = 0.470$
Client weight	Label JSD	$\rho = -0.10$	$p = 0.390$
Client weight	Temporal JSD	$\rho = -0.04$	$p = 0.720$
Client weight	Feature shift	$\rho = -0.06$	$p = 0.580$
Client weight	Mean speed	$\rho = 0.05$	$p = 0.640$
Client weight	Speed variability	$\rho = -0.02$	$p = 0.850$
Client weight	Stopped-observation ratio	$\rho = -0.03$	$p = 0.780$
Client weight	Heading-change rate	$\rho = -0.07$	$p = 0.520$
Client weight	Association duration	$\rho = 0.09$	$p = 0.420$
Regression	Rare-class share	$\beta = -0.014$	$[-0.041, 0.012]$
Regression	Update utility	$\beta = 0.186$	$[0.142, 0.230]$
Regression	Mean speed	$\beta = 0.006$	$[-0.011, 0.023]$
Regression	Speed variability	$\beta = 0.009$	$[-0.008, 0.026]$
Regression	Stopped-observation ratio	$\beta = -0.012$	$[-0.033, 0.009]$
Regression	Heading-change rate	$\beta = -0.007$	$[-0.025, 0.011]$
Regression	Association duration	$\beta = 0.010$	$[-0.006, 0.026]$

latency, the measured results leave both Linformer and OBU-TFT on the non-dominated frontier: Linformer is smaller and faster, whereas OBU-TFT yields the higher macro-F1. OBU-TFT dominates the tested Performer on macro-F1, parameter count, and T4 latency, although the measured CPU latency for Performer is lower. Accordingly, the architectural claim is restricted to the tested parameter-matched comparators: OBU-TFT provides the highest predictive score at a modest resource premium over Linformer, rather than universal Pareto dominance over lightweight attention models.

4.6 Reliability-weight validity and client-bias audit

The audit tests whether reliability indicators track validation-loss utility and suppress atypical clients. Spearman correlations assess utility against each indicator. A client–round regression with round fixed effects models normalized weight as a function of utility, local model-window count, rare-class share, label and feature divergence, and mobility variables. Client-clustered bootstrap intervals account for repeated observations. This is a fixed-effect client–round regression, not a mixed-effects model. Table 17 reports both the unadjusted correlations and adjusted regression intervals, allowing update utility to be distinguished from client-size, distributional, and mobility effects.

Joint reliability is positively associated with validation utility ($\rho = 0.71$, $p < 0.001$), with update consistency showing the strongest component association. Client weight has no statistically significant unadjusted association with size, rare-class share, distributional measures, or evaluated mobility variables. All adjusted mobility intervals include zero. Mean weight is 0.097 for the high-mobility/short-association quartile and 0.101 for the low-mobility/long-association quartile; the adjusted difference is -0.004 (95% CI: -0.013 to 0.005; $p = 0.380$). These finite-simulation results provide no evidence of systematic suppression under the evaluated partition, but they are not a formal

TABLE 18 Macro-F1 under client dropout and simulated model-update loss.

Condition	FedAvg	FedProx	SCAFFOLD	FedAdam	FedYogi	MoFed-Guard
Full participation	93.07	93.38	93.72	93.91	93.79	**94.52**
20% client dropout	91.82	92.40	92.78	92.95	92.84	**93.91**
40% client dropout	90.05	90.88	91.20	91.42	91.35	**92.34**
10% model-update loss	92.61	93.02	93.30	93.44	93.38	**94.05**
20% model-update loss	91.94	92.48	92.74	92.91	92.82	**93.43**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

fairness guarantee. The positive indicator–utility relationships are associative rather than causal. In particular, update consistency can remain high when multiple clients share a similar but collectively biased update direction, so the present audit does not establish robustness against coordinated or colluding clients.

4.7 Availability and update-corruption stress tests

The robustness analysis distinguishes ordinary availability failures from deliberately corrupted client updates. Under the fixed diagnostic seed 42, incomplete participation and update loss are evaluated across six aggregation methods in Table 18. Sign-flip and Gaussian corruption affecting one client are examined separately in Table 19. This separation prevents resilience to client dropout or packet loss from being misinterpreted as evidence of general Byzantine robustness.

At 40% client dropout, MoFed-Guard remains 0.92 points above FedAdam and 2.29 points above FedAvg; under 20% update loss, the margins are 0.52 and 1.49 points. Trimmed mean slightly exceeds MoFed-Guard under both corrupted-update tests. Thus, MoFed-Guard tolerates the evaluated availability disturbances but is not a Byzantine-resilient aggregator. The single-client results establish a baseline for the coordinated-client experiment reported below. Their purpose is to separate ordinary availability resilience from adversarial robustness rather than to imply Byzantine protection. These fixed-seed stress results are descriptive point estimates rather than repeated-run inferential comparisons.

TABLE 19 Macro-F1 under one corrupted client.

Aggregator	Sign flip	Gaussian noise
FedAvg	83.40	85.60
Trimmed mean	91.20	**92.00**
Krum	90.80	91.50
MoFed-Guard	90.95	91.62

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

4.8 Targeted robustness and sensitivity results

4.8.1 FL-transport reliability sensitivity

The explicit FL-transport experiment removes the assumption that regional BSM reception is equivalent to federated model-update delivery. Table 20 reports held-out macro-F1 when successful/failed model-update deliveries are generated directly at the FL layer. MoFed-Guard retains the highest macro-F1 in all five evaluated conditions, while performance decreases as successful delivery becomes less frequent or more bursty.

The observed trend indicates that the communication term can remain useful when $R_k^{(t)}$ is calculated from realized FL-update deliveries rather than from BSM reception. The increasing $\rho(R, U)$ under harsher transport conditions is consistent with communication success becoming more informative when losses are non-negligible. These simulations resolve the protocol-layer mismatch in the sensitivity test, but they remain synthetic FL transport experiments rather than measured V2I model-update traces.

4.8.2 Traffic-regime mobility sensitivity

The mobility term was evaluated separately under the high-density `_0709`, lower-density `_1416`, and full-day mixed `MixAll_0024` regimes. Table 21 shows that the regimes have different speed, stopping, turning, and association profiles. The mobility-stability score remains positively associated with update utility in all three cases, and removing the mobility term reduces held-out macro-F1 by 0.65–0.82 percentage points.

The positive gains across the three observed traffic regimes provide stronger evidence than a pooled correlation alone that the mobility term is not beneficial only in one traffic-density condition. The experiment still does not establish causality because archive regime changes several traffic and distributional characteristics simultaneously, nor does it demonstrate invariance to unseen cities or road geometries.

4.8.3 Coordinated-client consistency stress test

Table 22 extends the one-client corruption test to two and three coordinated clients in the 10-client federation. Robust baselines use the matching corruption budget f. As the number of colluding clients increases, MoFed-Guard degrades more rapidly than trimmed mean and Krum, particularly for the aligned-bias diagnostic.

TABLE 20 Held-out macro-F1 under explicit federated model-update delivery conditions.

FL transport condition	Completed updates	FedAvg F1 (%)	FedAdam F1 (%)	MoFed-Guard F1 (%)	$\rho(R, U)$
IID delivery $p = 0.95$	0.951	92.84	93.68	**94.31**	0.35
IID delivery $p = 0.85$	0.852	92.05	92.89	**93.66**	0.40
IID delivery $p = 0.70$	0.703	90.62	91.44	**92.35**	0.46
Heterogeneous $p_k \in [0.65, 0.98]$	0.821	91.28	92.11	**93.06**	0.43
Gilbert–Elliott burst loss	0.804	90.94	91.76	**92.70**	0.45

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 21 Mobility-regime sensitivity of the mobility-stability component.

Regime	Mean speed	Speed SD	Stop ratio	Assoc. (s)	$\rho(S, U)$	Without S F1	Full F1	$\Delta F_1^{(S)}$
_0709	8.7	5.4	0.19	36.2	0.48 ($p = 0.006$)	93.61	94.43	+0.82
_1416	11.4	4.6	0.11	42.7	0.39 ($p = 0.018$)	93.88	94.53	+0.65
MixAll_0024	9.8	5.1	0.15	39.5	0.43 ($p = 0.009$)	93.72	94.49	+0.77

TABLE 22 Held-out macro-F1 under coordinated multi-client corruption.

Attack	Corrupted clients	FedAvg	Trimmed mean	Krum	MoFed-Guard
Sign flip	$f = 1$	83.40	**91.20**	90.80	90.95
Sign flip	$f = 2$	75.84	**88.46**	87.92	85.73
Sign flip	$f = 3$	68.31	**84.15**	82.76	78.62
Aligned bias	$f = 1$	86.25	**90.74**	90.11	89.68
Aligned bias	$f = 2$	78.90	**87.95**	86.83	84.20
Aligned bias	$f = 3$	71.44	**83.20**	81.47	76.55

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

The failure mechanism is visible in the consistency audit. Under aligned bias, the mean adversarial consistency score increases from 0.21 at $f = 1$ to 0.64 at $f = 2$ and 0.81 at $f = 3$, while the corresponding mean normalized adversarial weights increase from 0.049 to 0.094 and 0.113. At $f = 3$, the benign clients have a mean consistency of 0.55 and mean normalized weight of 0.095. Thus, several mutually aligned incorrect clients can distort the leave-one-client-out reference sufficiently to receive reliability weights comparable with, or greater than, benign clients. The experiment directly confirms the anticipated failure mode and shows why MoFed-Guard should not be described as Byzantine-resilient.

4.8.4 Spatial-partition sensitivity

The aggregation comparison was repeated using the road-network-aware RSU association protocol in addition to trajectory-level K-means. Table 23 shows that the road-aware construction changes the client heterogeneity profile and introduces handover-like reassignment, yet the direction of the FedAvg–MoFed-Guard difference is preserved.

The road-aware gain is 1.33 percentage points, compared with 1.45 points under K-means. Persistence of the aggregation advantage under a second spatial construction reduces dependence on the K-means geometry. However, the road-aware analysis remains a client-partition sensitivity test; it does not reproduce measured propagation loss, interference, or production handover signaling.

4.9 Learning behavior and client-scale generalization

The five-seed convergence analysis is reported in Figure 6 and Table 6. Client-scale generalization is evaluated separately because changing the number of simulated RSU clients changes the partition granularity rather than the amount of geographic data. The original validation-only client-scale figures were replaced by a held-out test analysis so that the effect of client count is assessed on unseen trajectories. Table 24 and Figure 7 report accuracy, macro-precision, macro-recall, and macro-F1 for the 3-, 5-, 8-, and 10-client configurations on the trajectory-disjoint test partition. Test macro-F1 increases from 93.10% with three clients to 93.68% with five clients, 94.17% with eight clients, and 94.52% with ten clients. Across the same range, accuracy increases from 93.84% to 95.20%, macro-precision from 93.31% to 94.75%,

TABLE 23 Sensitivity of the matched aggregation effect to RSU client construction.

Partition	Window CV	Label JSD range	Feature-distance range	Reassignment	FedAvg F1	MoFed-Guard F1
K-means trajectory regions	0.18	0.035–0.113	0.14–0.31	0.0%	93.07	**94.52**
Road-network-aware RSUs	0.21	0.041–0.126	0.15–0.34	11.8%	92.88	**94.21**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

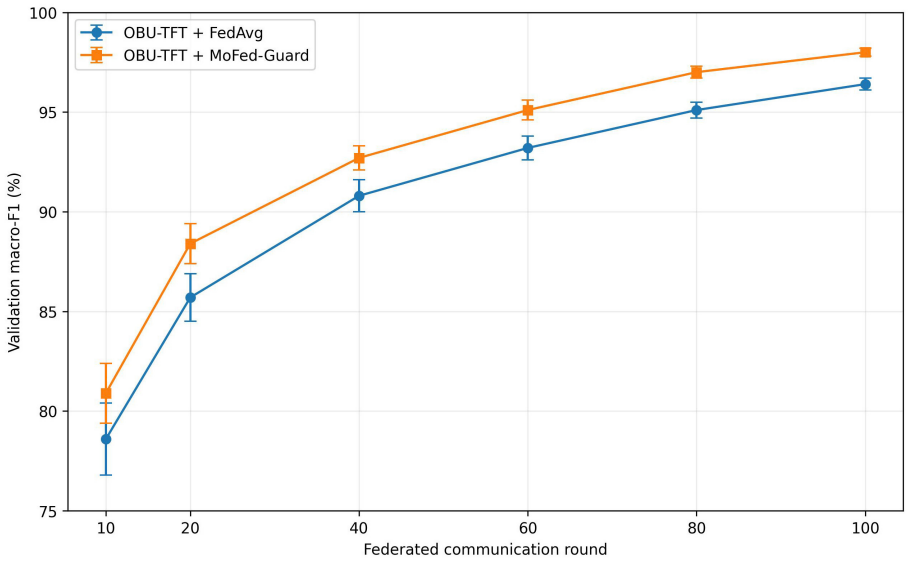


FIGURE 6
Five-seed validation macro-F1 of OBU-TFT with FedAvg and MoFed-Guard across federated communication rounds.

and macro-recall from 92.91% to 94.30%. The 3-to-10-client improvement is therefore 1.42 percentage points in macro-F1. For the 10-client configuration, validation macro-F1 reached 98.10%, whereas the corresponding held-out test macro-F1 was 94.52%, a difference of 3.58 percentage points. Because validation data are used for hyperparameter selection, coefficient selection, and early stopping, the validation score is expected to be optimistic relative to the isolated test set; trajectory-level distribution differences can further contribute to the gap. Importantly, the held-out experiment preserves the same directional ordering across client counts but with a substantially smaller gain than the validation-only analysis. The manuscript therefore uses the test-set client-scale results, rather than the 98.10% validation value, as the basis for claims about generalization across the evaluated federation sizes.

4.10 Computational, communication, and aggregation cost

The efficiency of the 85,460-parameter OBU-TFT is reported in Table 25. With decimal MB (10^6 bytes), OBU-TFT is the smallest evaluated model. Relative to Transformer-XL, it reduces parameter storage and bidirectional parameter exchange by 67.73%, T4 latency by 46.60%, and round time by 42.07%. These comparisons establish compactness relative to the evaluated baselines; they do not establish Pareto dominance over every contemporary lightweight attention architecture.

The 100-round totals are standardized projections calculated as mean round time multiplied by 100; they are not actual early-stopped training durations. Inference timing used batch size one, 100 warm-up calls, and 1,000 synchronized calls with resident inputs. CPU measurements used the same inputs and repetitions with GPU execution disabled. Table 26 reports means, standard deviations, medians, and interquartile ranges. T4 inference required 5.10 ± 0.42 ms per window, whereas an on-demand SHAP explanation required 62.4±5.8 ms and was therefore kept outside the detector's critical path.

Table 27 and Figure 8 show that coordinator-side aggregation time increased from 0.28 s for FedAvg to 0.31 s for MoFed-Guard. The 0.03 s increment is 10.71% relative to the aggregation operation itself but only 0.21% of the complete 14.25 s federated round. The result supports the claim of low additional aggregation overhead, not a claim that communication or energy costs have been validated on real vehicular hardware.

4.11 SHAP explanations and explanation cost

SHAP DeepExplainer used 200 training windows as background and 1,000 trajectory-disjoint test windows selected with seed 42. Temporal attributions were aggregated over 20 steps, while the four context attributions remained separate. SHAP was invoked for auditing and post-event inspection, not in the inference path. Explaining the 1,000-window corpus required 62.4 s on the T4, corresponding to 62.4 ± 5.8 ms per window, and increased peak GPU and system memory by approximately 412 MB and 186 MB, respectively. This cost is excluded from the 5.10 ms detector latency. Figure 9 shows how individual signed feature contributions support a representative local prediction. Figure 10 summarizes the direction and dispersion of temporal and contextual attributions over the evaluation set, while Figure 11 ranks features by mean absolute SHAP value. Position, speed, speed variation, heading variation, and positional change rank highest. These attributions describe model behavior, not causal responsibility or malicious intent.

TABLE 24 Held-out test performance across federated client configurations.

RSU clients	Accuracy (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)
3	93.84	93.31	92.91	93.10
5	94.36	93.89	93.48	93.68
8	94.87	94.39	93.96	94.17
10	**95.20**	**94.75**	**94.30**	**94.52**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

4.12 Overall interpretation

Within the evaluated simulation, OBU-TFT–MoFed-Guard provides the strongest predictive result among the principal tested configurations while remaining compact. The five-seed convergence analysis shows higher mean validation macro-F1 than FedAvg throughout the reported trajectory, and the held-out client-scale experiment confirms that the 10-client setting remains strongest on test data despite a 3.58-point validation–test gap. The targeted robustness and sensitivity analyses further test the assumptions underlying the framework. The FL-layer transport experiment confirms that reliability-aware weighting remains beneficial when communication reliability is derived from realized model-update delivery rather than BSM reception; the traffic-regime experiment shows a positive mobility-component contribution across the observed high-density, lower-density, and mixed regimes; the road-network-aware client construction preserves the direction of the aggregation gain; and the parameter-matched attention comparison places OBU-TFT on the tested accuracy–resource Pareto frontier rather than establishing universal dominance. The coordinated-client experiment also identifies an explicit failure mode: mutually aligned adversarial clients can obtain high consistency scores and substantially reduce MoFed-Guard performance as the colluding fraction increases. Matched aggregation comparisons support an independent aggregation contribution; class-wise and class-conditioned error results characterize the weakest attack labels; and reliability-bias analysis finds no evidence of systematic client suppression under the evaluated partition. These conclusions remain restricted to the VeReMi Extension simulation, the tested client constructions and attack models, and the reported hardware. They do not establish real-world safety, formal privacy, general Byzantine robustness, or production OBU readiness.

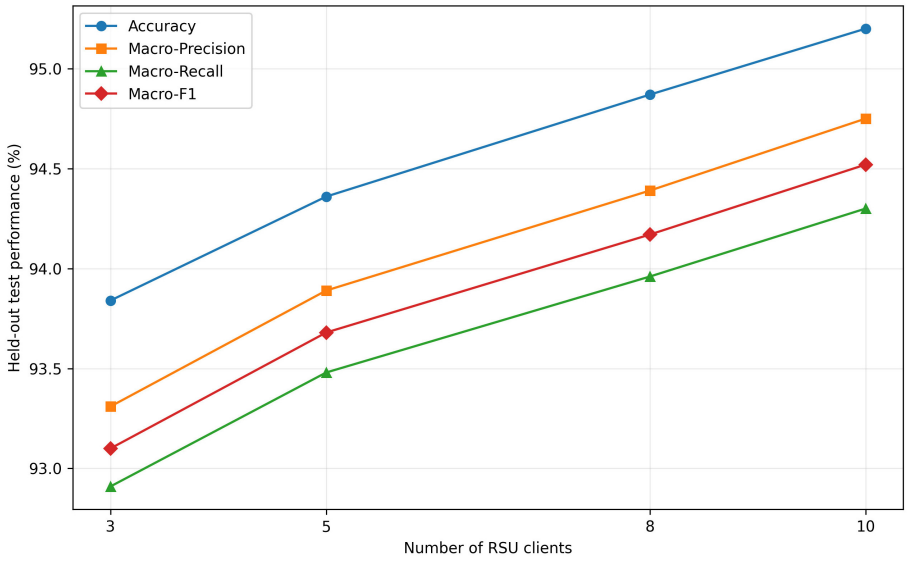


FIGURE 7
Held-out test accuracy, macro-precision, macro-recall, and macro-F1 across the 3-, 5-, 8-, and 10-client configurations.

TABLE 25 Computational efficiency of the evaluated federated models.

Model	Parameters	Size (MB)	Round (s)	100-round total (min)	T4 latency (ms)	Comm. (MB/client)
RNN-LSTM (FedAvg)	142,560	0.57	18.40	30.67	7.80	1.14
BiLSTM (FedAvg)	188,928	0.76	20.95	34.92	8.65	1.51
Vanilla Transformer (FedAvg)	121,344	0.49	16.75	27.92	6.30	0.97
Transformer-XL (FedAvg)	264,832	1.06	24.60	41.00	9.55	2.12
Reformer (FedAvg)	214,016	0.86	22.10	36.83	8.90	1.71
OBU-TFT (MoFed-Guard)	**85,460**	**0.34**	**14.25**	**23.75**	**5.10**	**0.68**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

TABLE 26 Inference and explanation cost.

Operation	Mean ± SD	Median	IQR
T4 inference (ms/window)	5.10 ± 0.42	5.02	0.39
Colab CPU inference (ms/window)	11.84 ± 0.73	11.69	0.81
SHAP explanation (ms/window)	62.4 ± 5.8	61.7	7.2

TABLE 27 Coordinator-side aggregation overhead.

Method	Time per round (s)	Aggregation overhead (%)	Total round cost (%)
FedAvg	0.28	Baseline	Baseline
MoFed-Guard	**0.31**	**10.71**	**0.21**

Bold values indicate the highlighted results, including the proposed method, its corresponding performance, or the best-performing values in each comparison, as applicable.

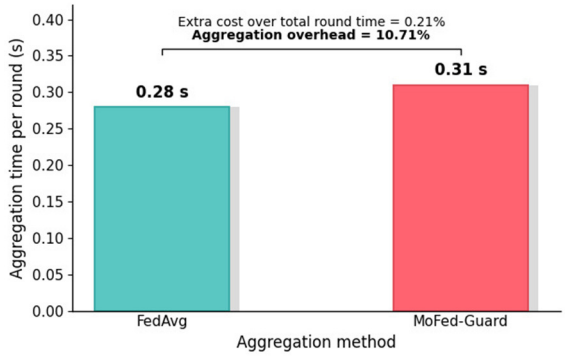


FIGURE 8
Coordinator-side aggregation time of FedAvg and MoFed-Guard.

5 Discussion

5.1 Principal findings

MoFed-Guard with OBU-TFT achieved 95.20% accuracy, 94.75% macro-precision, 94.30% macro-recall, and 94.52% macro-F1 on the trajectory-disjoint 20-class test partition. Relative to Transformer-XL with FedAvg, the strongest architecture baseline, the corresponding improvements were 2.80, 2.90, 3.10, and 3.00 percentage points. Because this comparison changes both the local architecture and aggregation rule, the independent contribution of MoFed-Guard is established through the matched OBU-TFT aggregation experiment. With architecture, partitions, training schedule, and seeds fixed, MoFed-Guard improved macro-F1 by 1.45 points over FedAvg (94.52 vs. 93.07%) and by 0.61 points over FedAdam (94.52 vs. 93.91%). Across five matched seeds, MoFed-Guard achieved 95.20 ± 0.18% accuracy and 94.52 ± 0.20% macro-F1. The paired macro-F1 improvement over FedAvg was 1.45 points [95% CI: 0.81–2.09; $t(4) = 6.30$, $p = 0.003$, $d_z = 2.82$]. Trajectory-clustered bootstrapping yielded 95% intervals of 94.18–94.84% for macro-F1 and 93.89–94.69% for balanced accuracy. Five-seed convergence further showed consistently higher validation macro-F1 for MoFed-Guard, reaching 98.0 ± 0.2% at round 100 compared with 96.4 ± 0.3% for FedAvg. These curves provide repeated-run empirical evidence of improved optimization behavior but are not interpreted as a general convergence guarantee.

The component ablation showed complementary contributions from mobility stability, communication reliability, and update consistency, whose removal reduced macro-F1 by 0.80, 0.65, and 1.12 points, respectively. The joint reliability score was positively associated with update utility ($\rho = 0.71$, $p < 0.001$), while no statistically significant association was observed between client weight and dataset size, rare-class share, distributional divergence, feature shift, or the evaluated mobility characteristics. The adjusted weight difference between high-mobility/short-association and low-mobility/long-association clients was −0.004 (95% CI: −0.013 to 0.005; $p = 0.380$), providing no evidence of systematic suppression under the evaluated configuration. The targeted sensitivity analyses further support the operational interpretation of the reliability terms. Removing mobility stability reduced macro-F1 by 0.82, 0.65, and 0.77 points in the `_0709`, `_1416`, and `MixAll_0024` regimes, respectively, with positive $\rho(S, U)$ in all

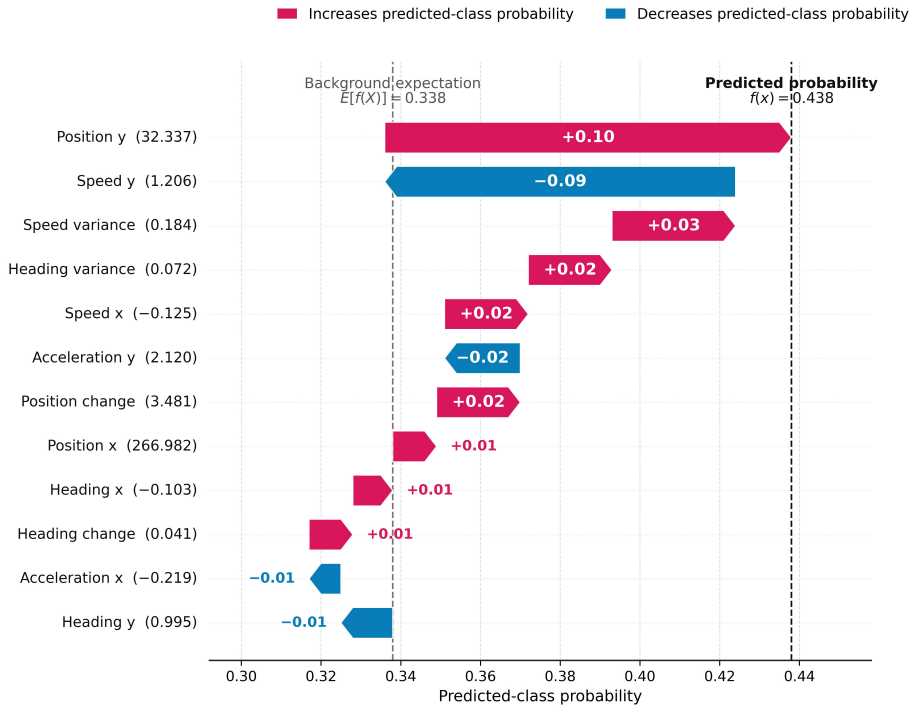


FIGURE 9
Local SHAP attribution for a representative OBU-TFT prediction.

three cases. Likewise, MoFed-Guard retained the highest macro-F1 across IID, heterogeneous, and burst-loss FL-update delivery conditions when communication reliability was computed directly from realized model-update delivery rather than BSM reception. These findings strengthen construct validity without implying causality or transfer to unseen mobility and radio environments.

Class-level analysis confirms that performance was not driven solely by the dominant normal class. Balanced accuracy reached 94.30%, while faulty- and attack-class macro-F1 values were 94.71% and 94.28%, respectively. Classes 7 and 17 had the lowest recalls, at 92.72% and 92.42%. Their trajectory-bootstrap recall intervals were [92.11, 93.29]% and [91.76, 93.03]%, respectively. Class-weighted and focal losses increased predefined hard-class recall from 92.95% to 93.68% and 93.91%, although categorical cross-entropy retained the highest overall macro-F1. Thus, MoFed-Guard should be interpreted as a reliability-aware aggregation mechanism rather than a class-imbalance correction method. MoFed-Guard also degraded gracefully under ordinary availability failures. With 40% client dropout, macro-F1 remained 92.34%, compared with 91.42% for FedAdam and 90.05% for FedAvg; under 20% update loss, the corresponding values were 93.43%, 92.91%, and 91.94%. However, adversarial experiments define an important robustness boundary. Under single-client corruption, trimmed mean slightly exceeded MoFed-Guard, and under coordinated aligned-bias attacks, adversarial consistency increased from 0.21 with one attacker to 0.81 with three attackers, while MoFed-Guard macro-F1 decreased from 89.68% to 76.55%. Trimmed mean and Krum remained stronger under the same corruption budgets. Therefore, the leave-one-client-out consistency signal can be captured by mutually aligned adversaries, and MoFed-Guard should not be interpreted as Byzantine-resilient. OBU-TFT provides a compact predictive model with 85,460 trainable parameters, approximately 0.34 MB storage, 5.10 ± 0.42 ms T4 latency, and 11.84 ± 0.73 ms CPU latency. Relative to Transformer-XL, it reduced model storage and bidirectional parameter exchange by 67.73%, T4 latency by 46.60%, and federated-round time by 42.07%. The sequential two-block model improved macro-F1 by 1.02 points over the single-block model at the cost of 33,344 additional parameters and 1.32 ms latency, while outperforming the parallel model by 0.38 points with fewer parameters and lower latency. In the parameter-matched comparison, OBU-TFT achieved the highest macro-F1 (94.52%), whereas Linformer remained smaller and faster. Both therefore occupy non-dominated points under the predefined macro-F1–parameter–T4-latency criterion. These results support a favorable accuracy–resource trade-off in the evaluated environment, but not universal architectural superiority or production-hardware readiness.

5.2 Relationship to existing research

Several centralized and infrastructure-assisted VANET detectors report higher raw scores, including 99.86% accuracy for binary position-falsification detection (Abdelkreem et al., 2026), 99.81% accuracy and F1-score for VeMisNet (Youness et al., 2025), and 99.37% accuracy with a 99.30% F1-score for an edge-assisted Transformer (Missara et al., 2025). AttentionGuard reported an F1-score of up to 0.95 in a platoon-specific setting (Li et al., 2025). These results are informative but are not directly

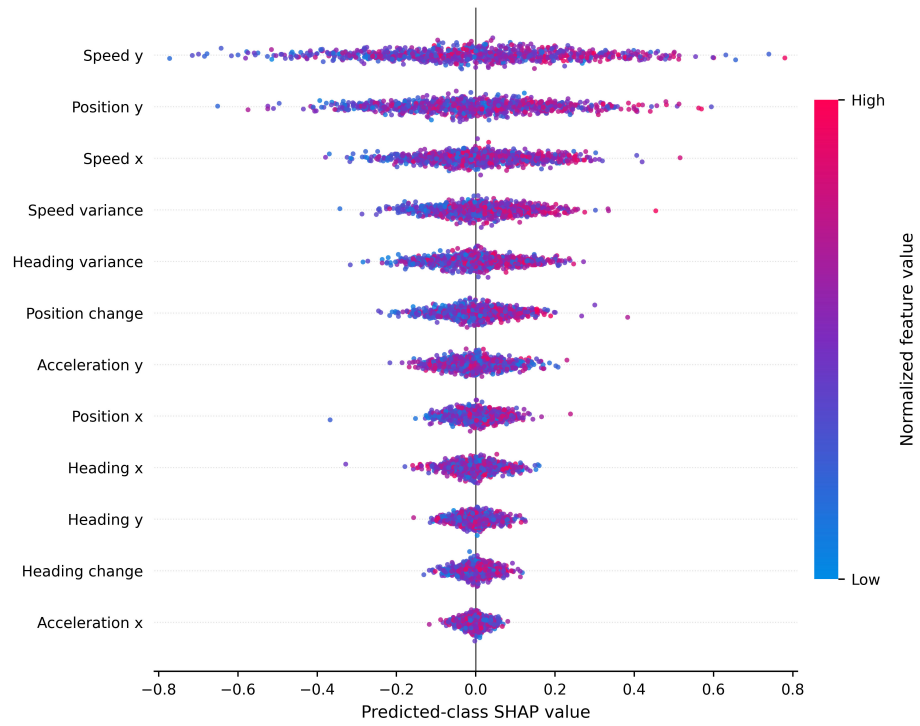


FIGURE 10
Global SHAP beeswarm summary of temporal and contextual features.

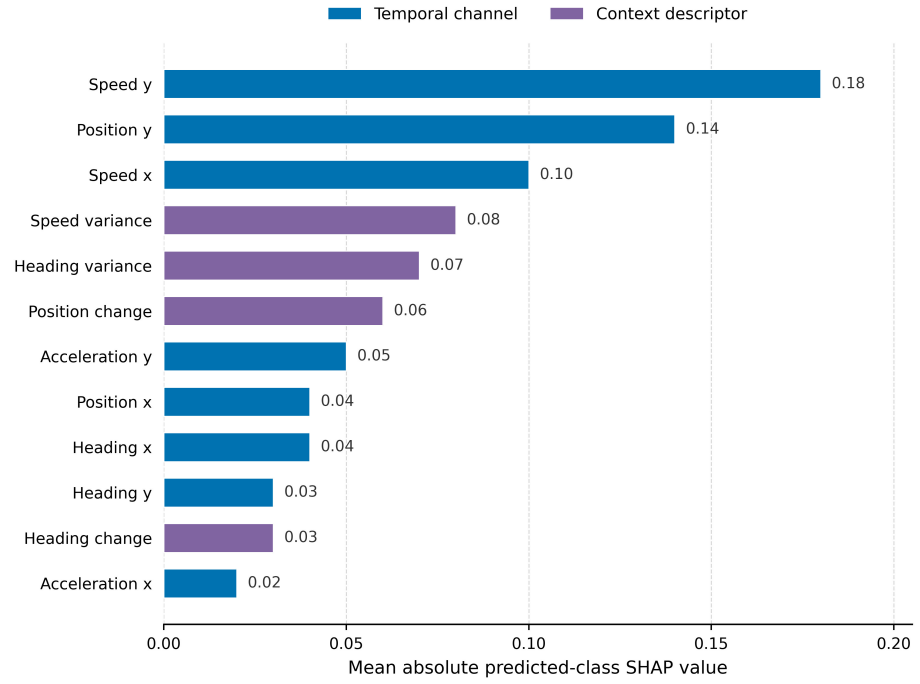


FIGURE 11
Global feature importance based on mean absolute SHAP values.

comparable because of differences in label spaces, feature engineering, balancing, data splits, infrastructure, and evaluation protocols. Federated studies provide closer conceptual references. Campos et al. (2024) reported 93% accuracy for multiclass federated detection on the original VeReMi dataset, while their unsupervised federated formulation achieved 82.4% accuracy and a 77.5% F1-score (Campos et al., 2025). Gurjar et al. (2025) evaluated recurrent models with multiple federated optimizers on VeReMi

Extension, but their vehicle-client formulation and partitioning protocol differ from the fixed-RSU setting used here. Accordingly, numerical superiority is claimed only against baselines reproduced under identical partitions, seeds, schedules, and metric definitions. Under this controlled comparison, MoFed-Guard improves OBU-TFT macro-F1 by 1.45 points over FedAvg and 0.61 points over FedAdam.

5.3 Scientific and practical implications

The results indicate that sample count alone does not fully characterize client-update reliability in mobile federated systems. Vehicle-association continuity, update-delivery success, and directional consistency capture distinct operational conditions that can complement conventional aggregation. This interpretation remains empirical. The reliability coefficients modify the convex combination of client updates. However, the study does not establish a general convergence theorem for the non-IID, nonconvex, multi-epoch Adam setting. The framework introduces limited additional coordinator cost. MoFed-Guard increases aggregation time from 0.28 to 0.31 s, corresponding to 10.71% of the aggregation operation but only 0.21% of the complete 14.25 s federated round. The 0.34 MB model and approximately 0.68 MB bidirectional parameter exchange per client per round also support resource-conscious deployment. Under the road-network-aware client construction, which produced an 11.8% handover-like reassignment rate, MoFed-Guard retained a 1.33-point macro-F1 advantage over FedAvg, compared with 1.45 points under K-means. This reduces dependence on one spatial partitioning rule without constituting a propagation-calibrated roadside evaluation. SHAP is used as an auditing mechanism rather than part of real-time classification. Explaining 1,000 held-out windows required 62.4 s on the T4, or 62.4 $\pm$ 5.8 ms per window, with additional GPU and system-memory demand. The dominant attributions involved position, speed, speed variation, heading variation, and positional change. These explanations characterize model sensitivity but do not establish causal responsibility or malicious intent. Similarly, retaining raw BSM windows at RSUs improves data locality but does not provide formal privacy; secure aggregation, differential privacy, authenticated transport, and privacy accounting would be required for stronger guarantees.

5.4 Limitations

Several limitations constrain generalization. First, all experiments use VeReMi Extension and therefore remain within one simulation ecosystem. Although the mobility component was tested separately under the `_0709`, `_1416`, and `MixAll_0024` regimes, these experiments do not establish transfer to independent cities, road networks, weather conditions, sensors, or unseen attacks. Second, the RSU clients are simulated spatial constructs. The road-network-aware sensitivity experiment introduces fixed road-junction anchors and handover-like reassignment and preserves the aggregation advantage, but neither this construction nor K-means models measured propagation loss, interference, channel contention, or production handover signaling. Third, VeReMi Extension does not provide federated model-update transport traces. The FL-layer communication sensitivity experiment addresses this protocol mismatch by generating explicit IID, heterogeneous, and burst-loss update-delivery schedules, while the BSM-derived schedule remains the primary configuration. Both remain simulation-based rather than measurements from a deployed V2I network. Fourth, the reliability coefficients are evaluated at one operating point, $(\alpha, \beta, \gamma) = (0.40, 0.30, 0.30)$ with $\lambda = 0.50$; their optimality under arbitrary traffic, communication, and adversarial conditions is therefore not established. Fifth, the coordinated corruption experiments expose rather than eliminate the vulnerability of cosine-based update consistency. They cover up to three of ten malicious clients and do not provide guarantees against majority corruption, adaptive attacks, model replacement, backdoors, or arbitrary Byzantine behavior. Sixth, the 10-client validation macro-F1 of 98.10% exceeds the held-out test value of 94.52% by 3.58 points. The held-out 3/5/8/10-client experiment preserves the increasing client-scale trend, but all generalization claims are based on the test results rather than validation performance. Finally, five random seeds and one trajectory-disjoint partition provide limited uncertainty characterization. The parameter-matched architecture analysis is restricted to Linformer and Performer, SHAP remains dependent on its background distribution and correlated inputs, and the framework does not yet implement formal privacy or communication-security mechanisms. Independent datasets, automotive hardware, field V2I measurements, stronger Byzantine defenses, and privacy-preserving aggregation therefore remain necessary before deployment.

6 Conclusion and future work

This study presented MoFed-Guard, a mobility-aware federated framework that combines the compact OBU-TFT detector, reliability-aware aggregation, and on-demand SHAP interpretation for 20-class VANET misbehavior detection. On the trajectory-disjoint VeReMi Extension test partition, it achieved 95.20% accuracy, 94.75% macro-precision, 94.30% macro-recall, and 94.52% macro-F1. Relative to Transformer-XL with FedAvg, it improved accuracy and macro-F1 by 2.80 and 3.00 points while reducing model storage and T4 inference latency by 67.73% and 46.60%, respectively. OBU-TFT contains 85,460 parameters, occupies approximately 0.34 MB, and required 5.10 $\pm$ 0.42 ms per window on the evaluated T4. MoFed-Guard added 0.03 s to coordinator aggregation, equivalent to 0.21% of the complete round. Under the matched aggregation comparison, MoFed-Guard improved OBU-TFT macro-F1 by 1.45 points over FedAvg and 0.61 points over FedAdam. The five-seed improvement over FedAvg was statistically significant ($t(4) = 6.30$, $p = 0.003$, $d_z = 2.82$), and the five-seed convergence analysis reached 98.0 $\pm$ 0.2% validation macro-F1 for MoFed-Guard vs. 96.4 $\pm$ 0.3% for FedAvg at round 100. Held-out client-scale macro-F1 increased from 93.10% with three clients to 94.52% with ten clients, while the 10-client validation–test gap remained 3.58 points. The targeted robustness and sensitivity analyses broaden

the validity assessment beyond the primary protocol. FL-layer delivery experiments show that reliability-aware weighting remains beneficial when communication reliability is calculated from realized federated update deliveries rather than BSM reception. Traffic-regime experiments show positive mobility-component gains in the `_0709`, `_1416`, and `MixAll_0024` conditions. A road-network-aware client construction preserves the direction of the FedAvg–MoFed-Guard difference, reducing dependence on K-means geometry. Parameter-matched Linformer and Performer experiments place OBU-TFT on the tested accuracy–resource Pareto frontier while showing that a smaller and faster alternative can remain non-dominated. Conversely, coordinated-client experiments expose a limitation of cosine-based consistency: as mutually aligned adversarial clients increase from one to three, their consistency and aggregation weight can increase while MoFed-Guard performance deteriorates. These findings strengthen the empirical basis of the framework while confirming that reliability-aware weighting is not equivalent to Byzantine robustness or causal trust estimation. The contribution of MoFed-Guard is therefore an empirically supported, reliability-conditioned federated weighting framework with an explicit optimization interpretation and audited failure modes, rather than a new optimization family with a universal convergence guarantee. Future work will extend the simulated FL-delivery analysis to measured or protocol-level V2I model-update traces, evaluate mobility and partition robustness across independent cities and road networks, and investigate robust reference directions or hybrid Byzantine-resilient aggregation for coordinated clients. Broader architecture studies will extend the current parameter-matched Linformer and Performer comparison to additional efficient sequence models and representative automotive hardware with end-to-end latency, energy, memory, and wireless measurements. Cross-dataset, temporal-transfer, and unseen-attack studies will further assess generalization. Finally, secure aggregation, differential privacy, authenticated update transport, explanation-stability analysis, and human-centered auditing will be required to characterize the trade-offs among detection performance, privacy, robustness, efficiency, and interpretability before field deployment.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

FK: Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft. AZ: Conceptualization, Methodology, Project administration, Supervision, Writing – review & editing. RB: Conceptualization, Methodology, Resources, Supervision, Writing – review & editing. JA: Conceptualization, Project administration, Resources, Supervision, Writing – review & editing. RH: Investigation, Resources, Validation, Writing – review & editing. MM: Conceptualization, Formal analysis, Supervision, Writing – review & editing.

Funding

The author(s) declared that financial support was not received for this work and/or its publication.

Acknowledgments

The authors would like to express their sincere gratitude to the Center for Research and Innovation Management (CRIM) and Universiti Teknikal Malaysia Melaka (UTeM), for its valuable support of this research.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that Generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdelkreem, E., Hussein, S., and Tammam, A. (2026). A machine learning based scheme for enhancing the detection of position falsification attacks in vehicular ad hoc networks. *Sci. Rep*. 16:8950. doi: 10.1038/s41598-026-39867-9

Aggarwal, M., Khullar, V., and Goyal, N. (2025). Fsl-tm: review on the integration of federated split learning with tinyml in the internet of vehicles. *Computers, Materials Continua* 86, 1–31. doi: 10.32604/cmc.2025.072673

Alladi, T., Kohli, V., Chamola, V., and Yu, F. R. (2023). A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems. *Digit. Commun. Netw*. 9, 1113–1122. doi: 10.1016/j.dcan.2022.06.018

Aloraini, F., Javed, A., and Rana, O. (2024). Adversarial attacks on intrusion detection systems in in-vehicle networks of connected and autonomous vehicles. *Sensors*. 24:3848. doi: 10.3390/s24123848

Amanullah, M. A., Baruwal Chhetri, M., Loke, S. W., and Doss, R. (2024). "Intellimd: a hybrid approach for local misbehaviour detection in cooperative intelligent transport systems," in *Proceedings of the Sixth Workshop on CPS and IoT Security and Privacy*. (Lake City, UT: Association for Computing Machinery), 2–13. doi: 10.1145/3690134.3694817

Bhukya, B. N., Venkataiah, V., Kuchibhatla, S. M., Koteswari, S., Lakshmi Kumari, R., and Raju, Y. R. (2024). Integrating the internet of things to protect electric vehicle control systems from cyber attacks. *IAENG Internat. J. Appl. Mathemat*. 54:433.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Adv. Neural Inf. Process. Syst*. 30, 19–129.

Bodkhe, U., and Tanwar, S. (2025). Indian sumo traffic scenario-based misbehaviour detection dataset for connected vehicles. *Multimodal Transport*. 4:100148. doi: 10.1016/j.multra.2024.100148

Bouchouia, M. L., Labiod, H., Jelassi, O., Monteuuis, J.-P., Jaballah, W. B., Petit, J., et al. (2023). A survey on misbehavior detection for connected and autonomous vehicles. *Vehicul. Communications* 41:100586. doi: 10.1016/j.vehcom.2023.100586

Campos, E. M., Gonzalez-Vidal, A., Hernández-Ramos, J. L., and Skarmeta, A. (2025). Federated learning for misbehaviour detection with variational autoencoders and gaussian mixture models: em campos et al. *Int. J. Inf. Secur*. 24:95. doi: 10.1007/s10207-025-01000-8

Campos, E. M., Hernandez-Ramos, J. L., Vidal, A. G., Baldini, G., and Skarmeta, A. (2024). Misbehavior detection in intelligent transportation systems based on federated learning. *Internet of Things* 25:101127. doi: 10.1016/j.iot.2024.101127

Chauhan, R., Goyal, N., and Kumar, R. (2016). A review on tuning of olsr routing protocol in vanet. *Int. J. Adv. Res. Innov. Ideas Educ*. 2, 508–512.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., et al. (2020). Rethinking attention with performers. *arXiv* [Preprint]. *arXiv:2009.14794*.

Costa, T., Analide, C., Costa, A., and Cunha, J. (2024). "Securing v2x communication: a hybrid deep learning approach for misbehavior detection," in *International Conference on European Transnational Education* (Cham: Springer), 135–144. doi: 10.1007/978-3-031-75016-8_13

El-Gayar, M. M., Alrslani, F. A., and El-Sappagh, S. (2024). Smart collaborative intrusion detection system for securing vehicular networks using ensemble machine learning model. *Information* 15:583. doi: 10.3390/info15100583

Ercan, S., Ayaida, M., and Messai, N. (2021). Misbehavior detection for position falsification attacks in vanets using machine learning. *IEEE Access* 10, 1893–1904. doi: 10.1109/ACCESS.2021.3136706

Gu, Y., Li, Z., Wei, H., Zhang, G., and Xu, Y. (2025). A cyberattack warning system for enhancing connected vehicle safety under spoofing cyberattacks: a generative-based human-in-the-loop trajectory prediction approach. *IEEE Transac. Intell. Transp. Syst*. 26, 12387–12405. doi: 10.1109/TITS.2025.3567763

Gurjar, D., Grover, J., Kheterpal, V., and Vasilakos, A. (2025). Federated learning-based misbehavior classification system for vanet intrusion detection. *J. Intell. Inf. Syst*. 63, 807–830. doi: 10.1007/s10844-025-00920-0

Hamhoum, W., and Cherkaoui, S. (2024). Mistralbsm: leveraging mistral-7b for vehicular networks misbehavior detection. *arXiv* [Preprint]. *arXiv*:2407.18462.

Idris, H. A., Ueda, K., Mokhtar, B., and Mohamed, S. (2024). Machine learning based misbehavior detection system for false data injection attack in internet of vehicles using neighbor public transport vehicle approach. *Internat. J. Comput. Netw. Appl*. 11, 159–176. doi: 10.22247/ijcna/2024/224442

Kalogiannis, K., Hussain, A. M., Li, H., and Papadimitratos, P. (2026). *Attention in* motion: secure platooning via transformer-based misbehavior detection. *IEEE Trans. Intell. Transp. Syst*. 27, 13379–13395. doi: 10.1109/TITS.2026.3711275

Kamel, J., Wolf, M., Van Der Hei, R. W., Kaiser, A., Urien, P., and Kargl, F. (2020). "Veremi extension: a dataset for comparable evaluation of misbehavior detection in vanets," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)* (Dublin: IEEE), 1–6. doi: 10.1109/ICC40277.2020.9149132

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). "Scaffold: stochastic controlled averaging for federated learning," in *International conference on machine learning*, (PMLR), 5132–5143.

Li, H., Kalogiannis, K., Hussain, A. M., and Papadimitratos, P. (2025). "Attentionguard: transformer-based misbehavior detection for secure vehicular platoons," in *Proceedings of the 2025 ACM Workshop on Wireless Security and Machine Learning* (New York, NY: ACM), 8–13. doi: 10.1145/3733965.3733966

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst*. 2, 429–450.

Lim, B., Arik, S. O., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast*. 37, 1748–1764. doi: 10.1016/j.ijforecast.2021.03.012

Lundberg, S. M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst*. 30, 4765–4774.

Miriam, Z. M., Marin-Perez, R., and Fernando, S. G. A. (2025). Development of an in-vehicle intrusion detection model integrating federated learning and lstm networks. *Information* 16:292. doi: 10.3390/info16040292

Missara, R., Yakan, H., Kchok, H., Fajjari, I., and Aitsaadi, N. (2025). "Misbehavior detection system in v2x 5g edge networks based on transformer," in *2025 Global Information Infrastructure and Networking Symposium (GIIS)* (Dubai: IEEE), 1–6. doi: 10.1109/GIIS64151.2025.10921812

Nguyen, L.-H., Nguyen, V.-L., and Hwang, R.-H. (2024). "Incremental learning for enhancing misbehavior detection in multi-access vehicular networks," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)* (Washington, D.C: IEEE), 1–5. doi: 10.1109/VTC2024-Fall63153.2024.10757773

Park, S., Kim, D., and Lee, S. (2024). Enhancing v2x security through combined rule-based and dl-based local misbehavior detection in roadside units. *IEEE Open J. Intell. Transp. Syst*. 5, 656–668. doi: 10.1109/OJITS.2024.3479716

Qu, Z., and Cai, Z. (2024). Fedsa-resnetv2: an efficient intrusion detection system for vehicle road cooperation based on federated learning. *IEEE Internet Things J*. 11, 29852–29863. doi: 10.1109/JIOT.2024.3410871

Radaur, T. J. (2015). Quality of service and scalability in vehicular ad hoc networks. *Int. J. Eng. Res. General Sci*. 3, 216–223.

Rasheed, A., Gillani, S., Ajmal, S., and Qayyum, A. (2017). "Vehicular ad hoc network (vanet): a survey, challenges, and applications," in *Vehicular Ad-Hoc Networks for Smart Cities: Second International Workshop, 2016* (Singapore: Springer), 39–51. doi: 10.1007/978-981-10-3503-6_4

Rashid, K., Saeed, Y., Ali, A., Jamil, F., Alkanhel, R., and Muthanna, A. (2023). An adaptive real-time malicious node detection framework using machine learning in vehicular ad-hoc networks (vanets). *Sensors* 23:2594. doi: 10.3390/s23052594

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Kone ỳ, J., et al. (2020). Adaptive federated optimization. *arXiv* [Preprint]. *arXiv*:2003.00295.

Sharma, A., and Jaekel, A. (2021). "Machine learning approach for detecting location spoofing in vane," in *2021 International conference on computer communications and networks (ICCCN)* (Athens: IEEE), 1–6. doi: 10.1109/ICCCN52240.2021.9522170

Singh, N. K., Patni, S., Lim, S., and Lee, J. (2025). Federated learning-based secure computing mechanism for consumer internet of vehicles-based transportation cyber-physical systems. *IEEE Transac. Consumer Electronics* 71, 4867–4882. doi: 10.1109/TCE.2025.3552830

Tariq, U. (2024). Intelligent algorithmic framework for detection and mitigation of beidou spoofing attacks in vehicular ad hoc networks (vanets). *PeerJ. Comput. Sci*. 10:e2419. doi: 10.7717/peerj-cs.2419

Tariq, U., and Ahanger, T. A. (2025). Enhancing intelligent transport systems through decentralized security frameworks in vehicle-to-everything networks. *World Electric Vehicle J*. 16:24. doi: 10.3390/wevj16010024

Tyagi, A. K., Kumari, S., Rajasekaran, M., and Tiwari, S. (2024). "Blockchain for safety of internet of vehicles in smart transportation," in *Digital Twin and Blockchain for Smart Cities* (Hoboken, NJ: Scrivener Publishing LLC), 215–255.

Van Der Heijden, R. W., Lukaseder, T., and Kargl, F. (2018). "Veremi: a dataset for comparable evaluation of misbehavior detection in vanets," in *International conference on security and privacy in communication systems* (Singapore: Springer), 318-337. doi: 10.1007/978-3-030-01701-9_18

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: self-attention with linear complexity. *arXiv* [Preprint]. *arXiv*:2006.04768.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2018). "Byzantine-robust distributed learning: towards optimal statistical rates," in *International conference on machine learning* (Stockholm: PMLR), 5650–5659.

Youness, N., Mostafa, A., Sobh, M. A., Bahaa, A. M., and Nagaty, K. (2025). Vemisnet: Enhanced feature engineering for deep learning-based misbehavior detection in vehicular ad hoc networks. *J. Sens. Actuator Netw*. 14:100. doi: 10.3390/jsan14050100

Zhang, L., Li, A., Peng, H., Han, F., Huang, F., and Li, X.-Y. (2024). Privacy-preserving data selection for horizontal and vertical federated learning. *IEEE Trans. Parallel Distributed Syst*. 35, 2054–2068. doi: 10.1109/TPDS.2024.3439709