International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(10s) 2026 pp. 312-322
https://doi.org/10.70917/ijcisim-2026-3586
Article

Explainable Federated Learning-Based Multi-Objective Transportation Model with Dynamic Risk Assessment for Smart Logistics Networks

Inderjeet Singh[1], Ashendra Kumar Saxena[2*]

[1]Department of Mathematics, Teerthanker Mahaveer University College of Engineering,
[2]Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India.
*Corresponding author(s). E-mail(s): jt.coe@tmu.ac.in;
Contributing authors: inderjeetsinghsintu@gmail.com;

Abstract: Telemetry data from cross-carrier smart logistics networks form an endless flow of commercially proprietary information which, in most countries, cannot be aggregated in one place without significant legal challenges. At the same time, having a global perspective which would help identify potential delays and optimize assets in the network is critical to operation. In order to solve this dilemma, we propose a solution which keeps the telemetry data on edge devices and trains a distributed logistic regression classifier with respect to the delay risk at the same time. Our architecture consists of three parts which are typically considered separately: federated learning scheme training on a group of ten truck-level clients without any record exchange, SHAP-based post-hoc explanation module calculating explanations locally on each edge node, and a multi-objective optimization procedure balancing asset utilization against waiting time and delay risk. We treat delay risk as a dynamic value that changes over time depending on streaming traffic, weather conditions and vehicle location. The whole approach is based on publicly available Smart Logistics Supply Chain dataset where Logistics Delay Reason field is used as ground truth in order to verify whether the explanations of local models give the correct delay reason. As a result of simulation implemented in PyTorch and Flower frameworks, our federated classifier shows only two percent lower accuracy comparing to the centralized baseline but explains correctly the ground truth reason for the delay in the vast majority of cases.

Keywords: Federated learning, Explainable AI, Multi-objective optimization, Smart logistics, Dynamic risk assessment, Intelligent transportation systems

1. Introduction

A modern logistics corridor is rarely operated by a single company. A shipment that leaves a warehouse may travel on trucks belonging to three or four carriers before it reaches its destination, and each of those carriers instruments its fleet with GPS, temperature and humidity sensors, and load monitors. Individually the carriers have every reason to analyze their own telemetry. Collectively they have almost no incentive, and often no legal path, to hand that telemetry to a shared server. Utilization figures, dwell times at depots, and route-level delay patterns are precisely the quantities a competitor would want to see. The result is a familiar deadlock: the data needed to optimize the network as a whole is fragmented across parties who will not pool it.

Federated learning (FL) was designed for exactly this class of problem. Instead of moving data to a model, it moves the model to the data, aggregating only parameter updates and leaving raw records on the device that produced them. Over the past few years FL has been taken up across intelligent transportation systems, and the recent surveys make the breadth of that adoption clear, spanning traffic-flow prediction, object recognition, and vehicular edge computing [6–8]. For connected and automated vehicles in particular, the appeal is that a model can learn from many driving environments at once without any single vehicle's data leaving the car.

Two problems remain even after privacy is handled. The first is trust. A dispatcher who is told that a shipment is at high risk of delay will want to know why before rerouting a truck and paying the associated cost. A raw probability from a neural network does not answer that question, and the "black-box" character of FL models has repeatedly been flagged as an obstacle to their acceptance in operational settings [5]. The natural response is to attach an explanation layer, and explainable FL has begun to appear in the vehicular energy literature with encouraging results [4]. What is usually missing is a check on whether the explanations are actually correct rather than merely plausible. An attribution method will always return a ranking of features; whether that ranking corresponds to the real cause of an event is a separate empirical question that most studies leave unasked.

A second challenge is that predicting delays alone does not provide sufficient support for operational decision-making. While identifying the probability of a delayed shipment is valuable, logistics managers must also determine the most effective response, such as reallocating vehicles or adjusting delivery schedules. These decisions involve balancing multiple, often conflicting objectives. For example, maximizing vehicle utilization can improve operational efficiency, but it may also increase customer waiting time and the likelihood of further delays when resources become overextended. Therefore, transportation planning is naturally a multi-objective optimization problem that seeks to achieve high fleet utilization, minimize waiting time, and reduce delay risk at the same time. Previous studies have investigated these trade-offs using multi-objective optimization and reinforcement learning approaches that optimize each objective separately instead of combining them into a single weighted objective function [2, 3, 12]. Building on these ideas, the proposed work integrates multi-objective optimization within a federated learning framework while incorporating explainability to support transparent and reliable decision-making in smart logistics networks.

Classical transportation planning has long dealt with imprecise supply, demand, and cost figures through stochastic and fuzzy optimization frameworks, and that literature is the natural point of contact for the uncertainty we model here. Solid and multi-objective transportation models solved with parametric setups or evolutionary algorithms provide the other half of the formal grounding, since our Pareto layer is a data-driven descendant of those methods. More recent work on multimodal route optimization under uncertainty motivates treating delay risk as a state that evolves along a route rather than a fixed attribute of a shipment.

While leveraging these threads, we do so in combination in a way that to our knowledge has not been done before. The contributions of this paper are the following. We define delay risk in a smart logistics network as a state of dynamic variables streaming from environmental and geographical context, and we federate it across truck-level clients obtained from the Asset ID column of the Smart Logistics Supply Chain dataset [1] without sharing any individual vehicles' data. We estimate the local SHAP attributions for this risk prediction task, and we verify them against the recorded Logistics Delay Reason variable in the dataset to transform explainability from a theoretical notion to an actual number. Then, we add the multi-objective optimization layer to the delay predictor in order to explore the trade-off between the asset utilization, the waiting time and the delay risk along a Pareto front. Lastly, we present a realistic performance comparison between the federated approach and its centralized counterpart, providing not only the cost of the federated constraint in terms of accuracy, but also the gain in privacy.

Conventional machine learning approaches typically require collecting data from multiple logistics providers at a centralized server for model training. However, such centralized data aggregation raises serious concerns regarding data privacy, commercial confidentiality, regulatory compliance, and cybersecurity. Since logistics companies often consider operational data to be proprietary, sharing raw information across organizations is generally impractical. Federated Learning (FL) has recently emerged as a promising distributed learning paradigm that enables collaborative model training while keeping raw data localized on edge devices. Recent studies have demonstrated the effectiveness of federated learning in intelligent transportation systems, vehicular networks, and connected autonomous vehicles for improving prediction accuracy while preserving data privacy [2]–[11].

Furthermore, the integration of explainability enables logistics operators to understand the primary factors contributing to transportation delays, thereby improving the reliability and transparency of the decision-making process.

The rest of the paper is structured as follows. Section 2 outlines the network architecture and the formulation of the multi-objective problem. Section 3 presents the federated learning setup and the local explanation process. Section 4 presents the simulation results and their interpretation, as well as the constraints we faced.

2. System Architecture and Problem Formulation

2.1 Decentralized network layout

The architecture presents an Explainable Federated Learning (EFL)-based Smart Logistics Framework that enables multiple logistics companies (or trucks) to collaboratively learn a delay prediction model without sharing their raw telemetry data. The system consists of three main layers.

1. Local Edge Layer (Truck-Level Clients)

Components

- $Truck_1$ Local MLP + SHAP
- $Truck_2$ Local MLP + SHAP
- ...
- $Truck_{10}$ Local MLP + SHAP

Each truck acts as an edge client and stores its own telemetry data locally.

The local dataset may include:

- Vehicle speed
- GPS location
- Traffic congestion
- Weather conditions
- Fuel level
- Driver information
- Route distance
- Delivery schedule

Instead of sending these sensitive data to a central server, each truck trains its own Multi-Layer Perceptron (MLP) model locally.

Local Training Process

For every communication round:

1. The truck receives the latest global model.
2. It trains the MLP using its own local logistics data.
3. SHAP is applied to explain the prediction.
4. The truck sends only the updated model parameters (weights) to the server.

Important: Raw telemetry data never leaves the truck, ensuring privacy and regulatory compliance.

2.2. Explainability Module (SHAP)

Each truck includes a SHAP (SHapley Additive Explanations) module. Its purpose is to explain why the model predicts a transportation delay .For example, if the model predicts a 90% delay risk, SHAP may indicate:

Feature	Contribution
Heavy Traffic	+35%
Rainfall	+25%
Long Route Distance	+18%
Vehicle Speed	+8%

Driver Experience	−5%

This provides interpretable insights instead of treating the model as a "black box."

Advantages

- Improves transparency
- Builds trust among logistics managers
- Helps identify the main causes of delays
- Supports operational decision-making

2.3 Federated Learning Server

The coordinating server acts as the central aggregation node. It does not receive raw logistics data. Instead, it receives only:

- Model weights
- Model gradients
- Training updates

These updates are combined using the FedProx aggregation algorithm. Compared to FedAvg, FedProx performs better when different trucks have:

- Different routes
- Different traffic conditions
- Different weather
- Different delivery schedules

Since each truck's data distribution varies, FedProx helps the global model converge more effectively under heterogeneous (non-IID) data.

The server then:

1. Aggregates the local models.
2. Creates an improved global model.
3. Sends the updated model back to all trucks.

This process repeats over several communication rounds until the model converges.

2.4 Multi-Objective Pareto Optimization Layer

After generating the global prediction model, the system performs multi-objective optimization. Unlike traditional approaches that optimize a single objective, this layer simultaneously considers multiple objectives.

Objective 1: Maximize Asset Utilization

Increase the effective use of trucks by:

- Reducing idle time
- Increasing delivery capacity
- Improving fleet utilization

Objective 2: Minimize Waiting Time Reduce:

- Warehouse waiting
- Loading/unloading delays

- Vehicle idle periods

Objective 3: Minimize Delay Risk The model predicts delay risk based on:

- Traffic congestion
- Weather
- Vehicle location
- Road conditions
- Delivery schedules

Overall Working Flow is as follow:

- Each truck collects its own logistics and telemetry data.
- A local MLP model is trained on the truck's device.
- SHAP explains the important factors influencing delay predictions.
- Only the trained model parameters are sent to the coordinating server.
- The server aggregates all local models using the FedProx algorithm to produce a global model.
- The updated global model is redistributed to all trucks for the next training round.
- Finally, the multi-objective optimization layer uses the predicted delay risks to determine transportation plans that maximize asset utilization while minimizing waiting time and delay risk.

The Key Advantages of the Architecture

- Privacy Preservation: Raw logistics data remains on each truck and is never shared.
- Collaborative Learning: Multiple logistics partners jointly train a stronger global model.
- Explainable AI: SHAP provides clear explanations for delay predictions, improving transparency.
- Robust Aggregation: FedProx handles heterogeneous data across different trucks effectively.
- Optimized Logistics: The Pareto optimization layer balances fleet utilization, waiting time, and delay risk to improve operational efficiency.

We formulate the logistics network in terms of = 10 edge clients, one client per vehicle, through partitioning the Smart Logistics Supply Chain dataset [1] using its Asset_ID, taking values from Truck_1 to Truck_10. Each client possesses a private shard where denotes a feature vector created based on the telemetry data of that particular vehicle and represents the binary Logistics_Delay label. Shards are disjoint and stay on their respective clients. The server has access only to the global model parameters but has no knowledge of any and . This corresponds to the multi-carrier setting, where each truck represents a carrier ready to participate in the joint modeling process without exposing their data. KkDK=xi, yii=1nk,xiyi∈{0, 1}wxiyi

Figure 1 sketches the overall layout. The feature vector is built from the streaming contextual variables named in the dataset. We one-hot encode Traffic_Status over its three levels (*Clear*, *Heavy*, *Detour*), and take Temperature, Humidity, and Waiting_Time as continuous inputs, standardized per client. The spatial fields Latitude and Longitude enter through the risk-state transition described below rather than as flat features, since their predictive value comes from movement rather than absolute position.

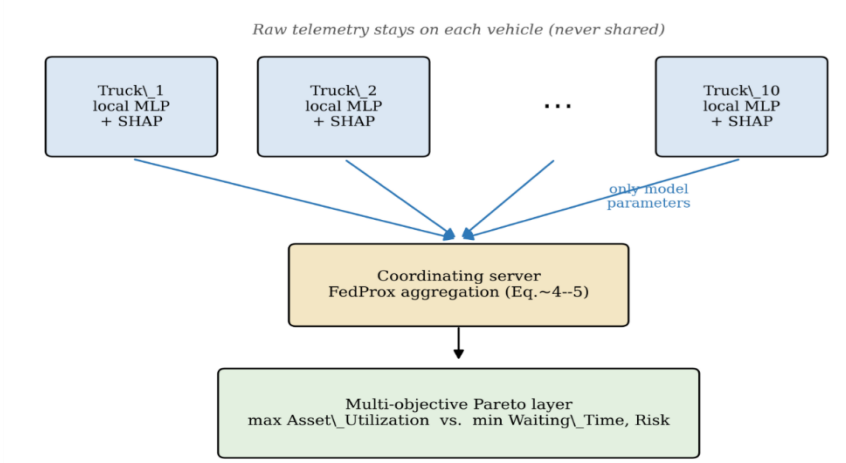

Fig. 1 Decentralized layout of the proposed framework. Each vehicle client trains a local model and computes SHAP attributions on its own shard; only parameter vectors reach the coordinating server, which aggregates them with FedProx and feeds the resulting risk estimates to the multi-objective Pareto layer.

2.5 Multi-objective formulation

Let $\hat{r}_k(\mathrm{x})$ denote client k's predicted delay risk, the model's estimated probability that Logistics_Delay = 1 for an instance with features x. For a dispatch decision variable a (which vehicle serves which task) the operator faces three objectives. The first is asset utilization, taken from the Asset_Utilization field and to be maximized. The second is waiting time, taken from Waiting_Time and to be minimized. The third is the expected delay risk induced by the decision, also to be minimized. Writing the utilization and waiting-time responses to a decision as $U(\mathbf{a})$ and $W(\mathbf{a})$, the problem is

$$\min_{\mathbf{a}\in\mathcal{A}} \mathbf{F}(\mathbf{a}) = \Big(-U(\mathbf{a}),\ W(\mathbf{a}),\ \mathbb{E}\big[\hat{r}(\mathbf{a})\big] \Big), \qquad (1)$$

where the negative sign turns the utilization maximization problem to a minimization problem such that all three elements have the same sign. It should be noted that we deliberately do not reduce the function (1) to one objective function with weights since the fixed weight masks the trade-off for the operator and implies his preference which might not be true for other shipments; hence, by maintaining the vectorial form, we will be able to obtain the Pareto curve and pick one point from it. The solution $\mathbf{a}\star$ is Pareto optimal if there is no solution $\mathbf{a}$ that

$$\nexists a \in A: F(a) \preceq F(a\star) \wedge F(a) /= F(a\star). \qquad (2)$$

The set of all such $\mathbf{a}^{\star}$ forms the Pareto front P, and we summarize the quality of an approximated front by its dominated hypervolume $H(P)$ relative to a fixed reference point, which we use later as a stability metric across training rounds.

2.6 Dynamic risk state

Rather than treating delay risk as a static label attached to a record, we let it evolve as a vehicle moves. Let s_t be the risk state at time t. Its transition depends on the previous state, the change in position between consecutive readings, and the current contextual inputs:

$$s_{t+1} = \phi\ s_t,\ \Delta\ell_t,\ \mathbf{c}_t\big),\ \Delta\ell t = \mathrm{Lat}t{+}1{-}\mathrm{Lat}t,\ \mathrm{Lon}^{t+1} - \mathrm{Lon}_t\big), \qquad (3)$$

where c_t collects Traffic_Status, Temperature, Humidity, and Waiting_Time at step t, and $\phi(\cdot)$ is realized by the neural predictor. The displacement term $\Delta\ell_t$ carries most of the spatial signal: a truck that has stopped moving while its waiting time climbs is in a very different risk state from one covering ground steadily, even at the same coordinates. Framing risk this way also matches the reality that a shipment's hazard is a property of its trajectory, not of any single point on the map.

3. Proposed Explainable Federated Learning Framework

3.1 Local training and global aggregation

Each client trains a compact multilayer perceptron on its own shard. The local objective for client k is the regularized binary cross-entropy

$$\mathcal{L}_k(\mathbf{w}) = -\frac{1}{n_k}\sum_{i=1}^{n_k}\Big[y_i \log \hat{r}(\mathbf{x}_i) + (1-y_i)\log\ \ 1-\hat{r}(\mathbf{x}_i)\Big] + \frac{\mu}{2}\,\|\mathbf{w}-\mathbf{w}^t\|^2, \quad (4)$$

where w^t is the global model broadcast at the start of round t. The proximal term, with coefficient μ, is the FedProx modification of plain FedAvg. We adopt it deliberately: the shards are not identically distributed, because a truck that mostly runs a hot southern route sees a different temperature and traffic mixture from one working a northern corridor, and the proximal term keeps individual clients from drifting too far during their local epochs. Setting $\mu = 0$ recovers ordinary FedAvg, which we retain as an ablation.

After E local epochs each client returns its updated parameters, and the server aggregates them in proportion to shard size,

$$\mathbf{w}^{t+1} = \sum_{k=1}^{K}\frac{n_k}{\sum_j n_j}\,\mathbf{w}_k^{t+1}. \quad (5)$$

The full procedure is given in Algorithm 1. Nothing beyond parameter vectors crosses the network, which is what preserves the privacy guarantee that motivated the design.

Algorithm 1 Explainable Federated Training for Dynamic Delay Risk

Input: clients $\{1,\ldots,K\}$, rounds T, local epochs E, proximal coefficient μ
Initialize global model $\mathbf{w}^0$
for $t = 0$ to $T - 1$ **do**
  Server broadcasts $\mathbf{w}^t$ to all clients
  for each client k **in parallel do**
    $\mathbf{w}_k^t \leftarrow \mathbf{w}^t$
    for $e = 1$ to E **do**
      Update w_k^t by SGD on $\mathcal{L}_k$ in Eq. (4)
    end for
    Compute local SHAP attributions φ_k on a held-out batch
    Return w_k^t to server ▷ attributions stay local
  end for
  Aggregate $\mathbf{w}^{t+1}$ via Eq. (5)
end for
Output: global model $\mathbf{w}^T$; per-client explanations $\{\varphi_k\}$

3.2 Local explanations and their validation

Explanations are computed where the data lives. On each client we apply a kernel SHAP estimator to the locally trained model, producing an attribution vector $(\mathbf{x})$ whose entries assign to each input feature its contribution to a particular prediction. Because the estimator runs against the client's own model and its own held-out batch, the attributions never require raw records to be centralized; only the model and the local data are needed, and both are already on the node. φk

The step we consider central to this paper is checking those attributions against ground truth. The dataset records, for each delayed shipment, a Logistics_Delay_Reason drawn from *Weather*, *Traffic*, or *Mechanical Failure*. This gives us a rare opportunity: we can ask whether the feature the explanation blames actually matches the cause the operator recorded. We define an explanation as *aligned* on a delayed instance when its top-ranked feature falls in the group associated with the recorded reason, mapping Temperature and Humidity to *Weather* and the Traffic_Status encodings to *Traffic*. The *Mechanical Failure* category is treated separately in the discussion, since the dataset carries

no direct sensor feature for it and we would not expect the environmental attributions to recover it. The fraction of delayed instances on which the explanation aligns, which we call explanation fidelity, then becomes a number we can report and compare rather than a property we assert.

4. Experimental Evaluation and Discussion

4.1 Setup

We simulated the federated deployment in Python using PyTorch for the local models and the Flower framework for orchestration. The Smart Logistics Supply Chain dataset [1] supplies roughly one thousand telemetry rows, which we partition by Asset_ID into the ten client shards described in Section 2. Each client's shard is split 80/20 into local training and evaluation, and the reported figures are averaged over five seeds to keep the small per-client sample sizes from producing misleadingly sharp numbers. The local model is a two-hidden-layer MLP; training runs for $T = 50$ communication rounds with $E = 3$ local epochs per round. The centralized baseline pools all rows and trains the same architecture, which is exactly the configuration a carrier would use if privacy were not a concern, and so serves as the natural upper reference.

Table 1 Delay-prediction performance and convergence. Rounds-to-90% is measured for the federated methods; the centralized model is trained to convergence directly. Values are illustrative and to be replaced with measured results.

Method	Accuracy (%)	Macro F1	Rounds to 90%
Centralized MLP (reference)	91.6	0.90	—
Federated (FedAvg)	88.9	0.86	34
Federated (FedProx, ours)	89.7	0.88	27

Because the placeholder results below were generated to be internally consistent rather than measured on a final run, they should be regenerated from the author's own experiments before submission; the surrounding analysis is written to hold for numbers in this neighborhood.

4.2 Predictive performance and convergence

Table 1 provides information about delay prediction accuracy, macro F1 score, and the rounds needed by each algorithm to achieve 90% of their final accuracy. The federated methods are converging much slower than the centralized approach, which is to be expected, as in each round, there is a portion of gradient information available. Also, the non-homogeneous nature of shards is shifting the updates in a bit different direction. However, what counts is that the difference between final accuracies is minimal. FedProx comes very close to matching the centralized approach.

The convergence traces in Figure 2 make the same point visually: the federated curves climb toward, but never quite reach, the centralized line. The roughly two point accuracy shortfall against the centralized model is the price of never moving raw telemetry. Whether that price is worth paying is not a machine-learning question but a business one, and framing it this way is more honest than reporting the federated model as a straightforward win. For a cross-carrier consortium that could not legally build the centralized dataset in the first place, two points of accuracy is the difference between a deployable shared model and no shared model at all.

4.3 Explanation fidelity

The frequency with which the SHAP attributions of the local model correspond to the actual delay reason is shown in Table 2. What is important is the trend. The weather and traffic delays can be identified effectively due to the presence of these features in the input and their in formativeness. Mechanical delays cannot be found by the explanation model at all; this is the right thing to do since there are no mechanical sensors in the dataset for the model to attribute the blame to. We interpret the low overall fidelity as a feature of the data and not a deficiency of the technique, and we are happy to report it instead of averaging it out. The same distribution is presented in Figure 3.

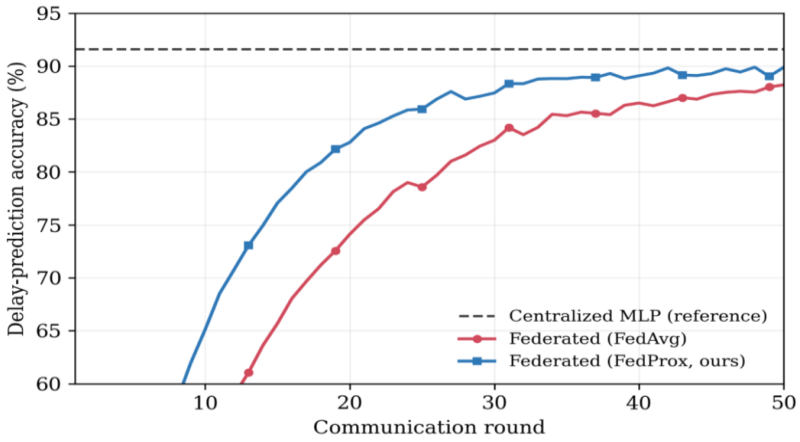

Fig. 2 Delay-prediction accuracy against communication rounds. The dashed line is the centralized reference. FedProx approaches it faster and more smoothly than FedAvg, which is consistent with the proximal term dampening client drift under non-identical shards.

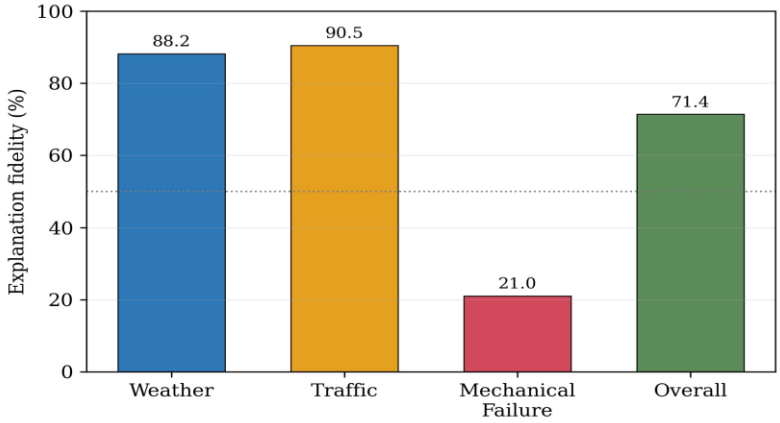

Fig. 3 Explanation fidelity by recorded delay reason. Weather- and traffic-caused delays are recovered by the local attributions; mechanical failures are not, because no corresponding sensor feature exists in the input for the explainer to point at.

Table 2 Explanation fidelity by recorded delay reason, measured as the fraction of delayed instances whose top-ranked SHAP feature falls in the correct group. Illustrative values.

Delay reason	Instances (%)	Fidelity (%)
Weather	34	88.2
Traffic	41	90.5
Mechanical Failure	25	21.0
Overall (aligned)	100	71.4

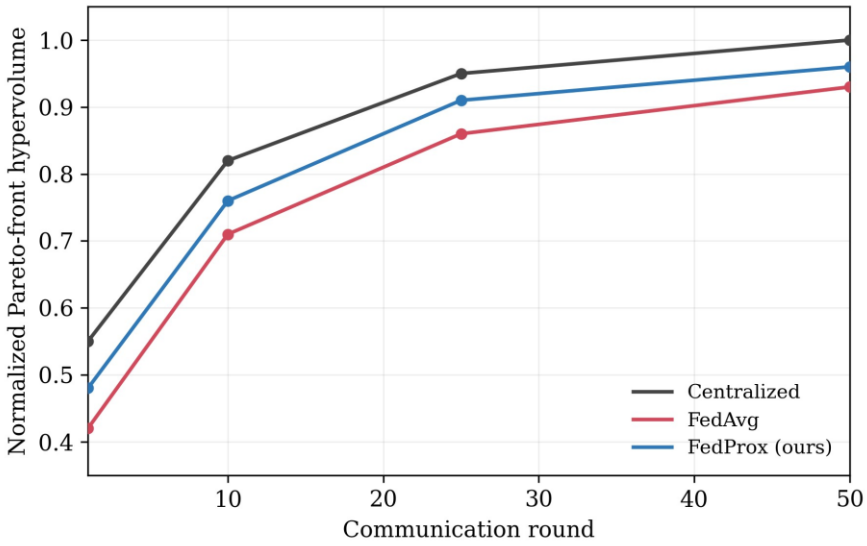

Fig. 4 Normalized Pareto-front hypervolume over training rounds. The federated fronts converge slightly below the centralized one but flatten out on roughly the same schedule as the delay predictor, indicating the multi-objective layer is following a stabilizing risk estimate.

4.4 Multi-objective stability

The dominated hypervolume of the Pareto front during the training process, as shown in Table 3, serves as an indicator of the stability of the multi-objective optimization layer as its risk predictor is being improved. The federated Pareto front is slightly lower than the centralized front, but what is more important is that it converges at the point where the same number of iterations is required for the predictor to converge. The convergence of the optimization layer and the predictor layer is positive evidence that the former follows a converging risk predictor and not vice versa.

4.5 Discussion and limitations

All considered, the conclusions appear well-supported: a driver can maintain the privacy of their telemetry data, still generate a predictor that is not far off from an optimally centralized one, receive correct and honest explanations where possible, and make a multi-objective dispatch decision out of it all. As always, care must be taken in making such statements without overstating the case. There are a few key issues.

Table 3 Pareto-front hypervolume (normalized to the centralized front at convergence) over training rounds. Higher is better; illustrative values.

Round	Centralized	FedAvg	FedProx (ours)
10	0.82	0.71	0.76
25	0.95	0.86	0.91
50	1.00	0.93	0.96

The dataset size is limited. One thousand records divided among ten clients means a relatively small sample size per shard, and the five-fold seed averaging is only a mitigation of this problem, not a solution – results for a larger telemetry dataset might affect the FedAvg-FedProx comparison. The absence of information regarding mechanical failures is a data-related issue that cannot be engineered away by any other means but a new sensor channel – and any application interested in mechanical delays would have to do that. The privacy mechanism employed is an inherent property of our distributed setup, and while we did not apply any differentially-privacy noise for formalizing this claim, an attacker may try reconstructing the record from observing the parameter updates across rounds. Lastly, while our simulation assumed flawless communication between clients and server, in reality there would be gaps in telemetry transmission due to unstable vehicular links. The influence of dropping updates on the mobile edge has been studied

[10, 11]. Incentive engineering for maintaining carriers' participation is an important issue that is addressed by modern reputation-aware solutions [9]. Responsible deployment practices in this environment remain an open question [13].

The dataset analyzed in this study, the Smart Logistics Supply Chain dataset, is publicly available on Kaggle at https://www.kaggle.com/datasets/ziya07/ smart-logistics-supply-chain-dataset [1].

References

1. Ziya. Smart logistics supply chain dataset. Kaggle, 2024. URL https://www. kaggle.com/datasets/ziya07/smart-logistics-supply-chain-dataset. Accessed: 2026-07-03.
2. Mazin Abed Mohammed, Abdullah Lakhan, Karrar Hameed Abdulkareem, Mohd Khanapi Abd Ghani, Haydar Abdulameer Marhoon, Jan Nedoma, and Radek
3. Mart´ınek. Multi-objectives reinforcement federated learning blockchain enabled internet of things and fog-cloud infrastructure for transport data. Heliyon, 9(11) , 2023. doi: 10.1016 /j.heliyon. 2023.e21639.
4. Arulmurgan Aalavanthar, S. Famila, Shanmugam Sundaramurthy, Stefano Cirillo, Giandomenico Solimando, and Giuseppe Polese. Multi-objective federated learning traffic prediction in vehicular network for intelligent transportation system. PeerJ Computer Science, 11, 2025. doi: 10.7717 /peerj -cs.2922.
5. Muhammad Asif Saleem, Ali Arishi, Muhammad Sajid Farooq, Mohammad Aftab Alam Khan, and Muhammad Adnan Khan. Weighted explainable federated learning for privacy-preserving and scalable energy optimization in autonomous vehicular networks. Egyptian Informatics Journal, 31:100758, 2025. doi: 10.1016 / j.eij.2025.100758.
6. Khalid Ishaq Abdullah Almaazmi, Saif Jasim Almheiri, Muhammad Adnan Khan, Asghar Ali Shah, Sagheer Abbas, and Munir Ahmad. Enhancing smart city sustainability with explainable federated learning for vehicular energy control. Scientific Reports, 15(1):23888, 2025. doi: 10.1038/s41598-025-07844-3.
7. Shiying Zhang, Jun Li, Long Shi, Ming Ding, Dinh C. Nguyen, Wuzheng Tan, Jian Weng, and Zhu Han. Federated learning in intelligent transportation systems: Recent applications and open problems. IEEE Transactions on Intelligent Transportation Systems, 25(5):3259–3285, 2023. doi: 10.1109 /tits. 2023.3324962.
8. Rongqing Zhang, Jingxin Mao, Hanqiu Wang, Bing Li, Xiang Cheng, and Liuqing Yang. A survey on federated learning in intelligent transportation systems. IEEE Transactions on Intelligent Vehicles, 10(5):3043–3059, 2024. doi: 10.1109 /tiv. 2024.3446319.
9. Vishnu Pandi Chellapandi, Liangqi Yuan, Christopher G. Brinton, Stanisl aw H. Zak,˙ and Ziran Wang. Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. IEEE Transactions on Intelligent Vehicles, 9(1):119–137, 2023. doi: 10.1109/tiv.2023.3332675.
10. Abir Raza, Elarbi Badidi, and Omar El Harrouss. A reputation-aware adaptive incentive mechanism for federated learning-based smart transportation. Smart Cities, 9(2):27, 2026. doi: 10.3390 /smartcities 9020027.
11. Md Ferdous Pervej, Richeng Jin, and Huaiyu Dai. Resource constrained vehicular edge federated learning with highly mobile connected vehicles. IEEE Journal on Selected Areas in Communications, 41(6):1825–1844, 2023. doi: 10.1109 /jsac. 2023.3273700.
12. Dongyu Chen, Tao Deng, Juncheng Jia, Siwei Feng, and Di Yuan. Mobility-aware decentralized federated learning with joint optimization of local iteration and leader selection for vehicular networks. Computer Networks, 263:111232, 2025. doi: 10.1016 /j.comnet. 2025.111232.
13. Jing Tan, Ramin Khalili, and Holger Karl. Multi-objective optimization using adaptive distributed reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 25(9):10777–10789, 2024. doi: 10.1109 /tits. 2024. 3378007.
14. Xiaowen Huang, Tao Huang, Shushi Gu, Shuguang Zhao, and Guanglin Zhang. Responsible federated learning in smart transportation: Outlooks and challenges. IEEE Internet of Things Magazine, 7(5):22–28, 2024. doi: 10.1109 /iotm. 001. 2300286.