Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 3s (2025)

# Nonlinear Decentralized Federated Localization Framework for Dynamic Vehicular Networks

***[1]B. Deena Divya Nayomi, [2]V. Jyothsna**

[1]Research Scholar, Department of Computer Science and Engineering School of Computing, MOHAN BABU UNIVERSITY (Erstwhile Sree Vidyanikethan Engineering College (Autonomous)), Tirupati, Andhra Pradesh, India.

Email: bdeenadivyanayomi@gmail.com

[2]Associate Dean (Academic Affairs) and Associate Professor, Department of Data Science School of Computing, MOHAN BABU UNIVERSITY

(Erstwhile Sree Vidyanikethan Engineering College), Tirupati, Andhra Pradesh, India.

Email: jyothsna1684@gmail.com

***Article History:***

***Received**: 21-09-2024*

***Revised**: 30-10-2024*

***Accepted**: 20-11-2024*

**Abstract:**
This study introduces a Nonlinear Decentralized Federated Localization Framework (DFLF) designed for precise real-time vehicle positioning in dynamic vehicular networks. The framework addresses GNSS inaccuracies, high communication costs, and privacy challenges by integrating nonlinear machine learning techniques, including Long Short-Term Memory (LSTM) networks, Graph Convolutional Networks (GCNs), and transformer-based attention mechanisms. These models effectively capture complex temporal and spatial dependencies from GNSS, sensor, and vehicle-to-vehicle (V2V) communication data. Federated learning ensures secure, decentralized training by exchanging encrypted model updates rather than raw data. Experimental results using synthetic GNSS data and V2V interactions demonstrate significant accuracy improvements, with root mean square errors (RMSE) of 2.3 meters in urban scenarios and 1.2 meters on highways. Scalability tests with networks of up to 500 vehicles confirm the model's robustness in dense traffic environments. Privacy-preserving measures such as differential privacy and homomorphic encryption ensure secure collaboration without notable performance degradation. The framework's nonlinear modeling capability enhances localization reliability under urban canyon conditions and intermittent connectivity, making it a viable solution for decentralized vehicle positioning in real-world scenarios.

**Keywords**: Federated Localization, GNSS Accuracy, Vehicular Networks, Data Privacy, Hybrid Learning, LSTM, GCN, V2V Communication.

## 1 Introduction

In vehicular networks, accurate node positioning is crucial for traffic safety, navigation, and coordination. Communication overhead, privacy risks, and challenges in handling dynamic environments often limit centralized localization methods, where data frequently changes. Decentralized federated learning (DFL) offers an alternative solution by enabling vehicles to train shared models without sending raw data to a central server [1], [2]. Privacy concerns are reduced, and centralized infrastructure requirements are minimized, particularly in scenarios involving sensitive location data. Vehicles share model updates directly instead of raw information, facilitated by DFL.

Diverse data sources enhance learning through weighted aggregation methods like the DFL-DDS algorithm, even when vehicles are constrained by limited communication ranges or specific routes [2]. By sharing and validating positional data, cooperative localization methods, including Local Dynamic Maps (LDMs), enhance accuracy [3]. Without centralized aggregation, federated filtering and consensus optimization techniques integrate data from multiple sources, further improving localization [4], [5].

Ensuring accurate model convergence and efficient communication in highly dynamic vehicular settings remains a significant challenge. Network topology and traffic density frequently change, reducing model accuracy and slowing convergence [6]. Bandwidth usage must be reduced while maintaining the integrity of shared updates, which requires efficient methods like adaptive gradient compression [1][6]. Addressing these gaps and improving localization reliability necessitate exploring the integration of additional data sources, such as GPS and UWB [5]. To address these issues, a Decentralized Federated Localization Framework (DFLF) is proposed in this research. Through decentralized learning models, the study aims to improve localization accuracy, reduce communication costs using optimized update strategies, and maintain data privacy by applying cryptographic techniques during model training. The framework enables collaboration among vehicles while preserving sensitive information and focuses on combining temporal and spatial data from multiple sensors. The reliability of localization in dynamic vehicular networks is expected to improve with the proposed framework. Applications of this framework include supporting autonomous vehicle systems, traffic coordination, and real-time navigation. Experimental tests using datasets like KITTI and real-world vehicular network data will validate the framework's performance [7], [5].

The paper is structured as follows: A detailed explanation of the framework components, including data sources, preprocessing steps, and learning methods, is presented in Methods and Materials. Next, experimental results and their analysis are discussed, followed by conclusions and suggestions for future work.

## 2 Related work

Ashour et al. [4], Wang et al. [5], Parker et al. [8], Pollicino et al., [9] and Leung et al. [10] focus on decentralizing vehicle location systems. Ashour et al. [4] describe a system that reduces location errors by 62.8% by allowing vehicles to share their position data. Wang et al. [5] explain a vehicle-to-vehicle (V2V) system that estimates positions with 20–30 cm accuracy but still needs testing in large, busy areas. Parker et al. [8] suggest using data shared directly between vehicles for better real-time location estimates. However, practical tests or detailed data are missing. Leung et al. [10] add to this idea by adapting their method to low-traffic areas but do not explain how it scales to busier regions.

Li et al. [11], Movahedian et al. [12], Zhou et al. [7], and Su et al. [2] explore federated learning for cars. Li et al. [11] describe a system (C-DFL) that keeps personal data private while improving learning accuracy. Resource use during distributed training and its limits are not fully explained. Movahedian et al. [12] describe how model aggregation can save up to 83% of computing resources, though it is unclear how it works with fast-changing networks. Zhou et al. [7] combine time and space data to improve learning transfer but mention possible issues with processing delays. Su et al. [2] suggest using varied data sources to improve accuracy while keeping privacy. Still, how it handles heavy communication loads is not detailed.

Fu et al. [13], Santos et al. [14], and Lee et al. [15] examine roadside units (RSUs) and mixed networks. Fu et al. [13] describe a system that tracks vehicles using RSUs, cutting GPS errors from 7.21 m to 0.74 m. Santos et al. [14] also use RSUs and show similar improvements in positioning. Both papers skip discussing costs or how RSUs would work in larger areas. Lee et al. [15] combine different types of networks to improve position accuracy but do not explain how the networks interact or deal with scaling issues.

Al-Hattab et al. [16] and Mohammadabadi et al. [17] focus on mathematical methods for finding locations. Al-Hattab et al. [16] use signal strength and distance to find sensor locations. The ideas are theoretical, with no practical testing mentioned. Mohammadabadi et al. [17] use cooperative positioning to improve accuracy by up to 50%. Practical use in busy areas or traffic scenarios is not explained well.

Barbieri et al. [1] and Kong et al. [18] focus on lowering the amount of data exchanged in networks. Barbieri et al. [1] describe compressing data to use fewer resources. Tests in crowded traffic situations are not included. Kong et al. [18] introduce FedVCP, which reduces GPS errors but does not address possible delays or communication problems in heavy traffic.

The papers show how decentralized and federated systems may help with positioning and data sharing in networks. Many methods rely on sharing data while maintaining privacy and reducing central dependencies. However, testing in real-world situations, managing communication loads, and handling costs or scaling challenges are often not discussed fully.

## 3 Methods and materials

### 3.1 Framework Overview

The Decentralized Federated Localization Framework (DFLF) organizes data from vehicles to calculate positions more accurately in changing environments. It gathers information from GNSS, vehicle-to-vehicle communication using IEEE 802.11p, and onboard sensors like accelerometers and gyroscopes. Each data source contributes unique details about the vehicle's location and movement.

The framework works through five layers. The data acquisition layer collects information in real time, including coordinates from GNSS, distance and speed data from V2V communication, and movement-related measurements from onboard sensors. Noise and inconsistencies in the data are handled in the edge computing layer. Filtering techniques like Kalman filters smooth out the input, and useful details like speed and direction are extracted to prepare the data for analysis.

The communication layer connects vehicles directly, allowing them to exchange information about their trained models instead of raw data. This layer uses protocols like IEEE 802.11p to keep the data flow steady, even when vehicles join or leave the network. The federated aggregation layer combines these updates from multiple vehicles, calculating shared improvements to the positioning models. The method relies on weighted averages to balance contributions from vehicles with different amounts or quality of data.

The final layer uses machine learning to estimate positions. It combines patterns in time from GNSS and sensor data with spatial relationships from vehicle interaction graphs. The analysis blends techniques like LSTMs for handling sequences and GCNs for mapping connections between vehicles. A transformer-based approach integrates these inputs to provide accurate positioning predictions.

This setup reduces dependency on central servers and adapts to changing traffic patterns. A diagram showing how data moves between layers can help visualize this process.

### 3.2 Data Sources

The framework collects data from three main sources to estimate vehicle positions and movements. GNSS provides geographic information such as latitude and longitude. This data forms the basis for locating vehicles but often encounters accuracy issues in urban areas. Signals may bounce off tall buildings or become blocked by obstacles, leading to errors, especially in places like tunnels or dense city centers.

Onboard sensors add extra details about a vehicle's motion. Accelerometers measure changes in speed, while gyroscopes detect shifts in direction. Odometers track the distance traveled, helping to calculate the vehicle's position through a method called dead reckoning. Cameras, when included, detect surrounding features like road markings or nearby objects, offering additional context to improve position estimates.

IEEE 802.11p communication allows vehicles to exchange information with one another. This includes details such as the distance between vehicles and their relative speeds. By sharing this data, vehicles gain a better understanding of their surroundings. This type of communication becomes especially useful in situations where GNSS data or sensors alone cannot provide enough accuracy. Combining these three data sources improves the overall quality and reliability of localization in different driving environments.

### 3.3 Data Preprocessing

Preprocessing begins with handling noise in the raw data. GNSS signals and sensor readings often include inconsistencies caused by environmental interference or equipment limitations. Noise is reduced using Kalman filters, which predict and correct positions iteratively. The prediction step calculates the next position based on the current state:

$$x_{k+1} = Ax_k + Bu_k$$

Here, $x_k$ represents the vehicle's current state, $A$ is the state transition matrix, and $u_k$ is the control input. The update step adjusts this prediction using new sensor measurements:

$$x_{k+1}{}' = x_{k+1} + K(z_{k+1} - Hx_{k+1})$$

The term $z_{k+1}$ is the actual measurement, $H$ maps the predicted state to measurement space, and $K$ is the Kalman gain, calculated as:

$$K = PH^T(HPH^T + R)^{-1}$$

where $P$ is the error covariance matrix, and $R$ represents measurement noise. This process reduces random fluctuations while refining position estimates.

For cases involving nonlinear patterns, Particle filters are applied. A set of weighted particles approximates the probability distribution of the vehicle's state. Particles are propagated through the system's motion model, and their weights are updated based on sensor likelihoods. The state is estimated as the weighted average of these particles.

After noise reduction, features are extracted to prepare data for further analysis. Velocity is calculated from position changes over time using:

$$v = \frac{\Delta x}{\Delta t}$$

Acceleration is derived from changes in velocity:

$$a = \frac{\Delta v}{\Delta t}$$

Time-series patterns are identified by examining GNSS data trends over successive intervals. Vehicle-to-vehicle communication data, such as distances, are represented in graph structures. Each vehicle acts as a node, and the distance between vehicles forms the edges, creating a spatial representation for later use.

1. **Input:** Raw GNSS data $(z_k)$, sensor data $(u_k)$, V2V distance measurements.
2. **Noise Reduction:**
   - Predict the state: $x_{k+1} = Ax_k + Bu_k$
   - Update the state: $x_{k+1}{}' = x_{k+1} + K(z_{k+1} - Hx_{k+1})$
   - Compute Kalman gain: $K = PH^T(HPH^T + R)^{-1}$
   - For nonlinear cases, propagate particles and adjust weights.
3. **Feature Extraction:**
   - Compute velocity: $v = \frac{\Delta x}{\Delta t}$
   - Compute acceleration: $a = \frac{\Delta v}{\Delta t}$
   - Construct graphs from V2V data: Nodes represent vehicles, and edges represent distances.
4. **Output:** Preprocessed data including refined positions, motion features, and spatial graphs.

This preprocessing workflow ensures data consistency and prepares it for input into machine learning models. It combines mathematical operations and structured techniques to make data usable for later stages.

### 3.4 Machine Learning Models

The proposed hybrid model combines methods to analyze temporal, spatial, and multi-source vehicular data. Long Short-Term Memory (LSTM) networks process time-series patterns, while Graph Convolutional Networks (GCNs) examine spatial relationships. A transformer-based attention mechanism combines these features. Federated Averaging (FedAvg) allows decentralized learning, where nodes share model updates instead of raw data.

LSTM networks capture dependencies in sequences like GNSS data. The model predicts and updates the hidden and cell states using the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Here, $f_t$, $i_t$, and $o_t$ represent the forget, input, and output gates, $C_t$ is the cell state, and $h_t$ is the hidden state at time $t$. These equations help to track time-based trends in vehicle motion.

GCNs map spatial relationships in data collected from vehicle interactions. Vehicles are treated as nodes, and distances between them form weighted edges in a graph. The GCN layer propagates information through nodes as:

$$H^{(k+1)} = \sigma(\widetilde{D}^{-1/2} \tilde{A} \widetilde{D}^{-1/2} H^{(k)} W^{(k)})$$

In this formula, $\tilde{A}$ represents the adjacency matrix with added self-loops, $\widetilde{D}$ is the degree matrix, $H^{(k)}$ contains node features at layer $k$, and $W^{(k)}$ is the weight matrix. This calculation updates each node's features by aggregating information from its neighbors.

The transformer-based attention mechanism combines temporal and spatial features by assigning weights to important data points. Attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Here, $Q$, $K$, and $V$ are query, key, and value matrices derived from input features, and $d_k$ is the dimension of the key. This process identifies which features contribute most to position predictions.

Federated Averaging enables decentralized learning by aggregating local models from multiple nodes. The global model is updated as:

$$w_t = \frac{\sum_{i=1}^{N} n_i w_t^i}{\sum_{i=1}^{N} n_i}$$

where $w_t^i$ represents the parameters of the model trained at node $i$, $n_i$ is the data size at that node, and $w_t$ is the updated global model at iteration $t$.

### 3.4.1 Algorithm Model-Based Hybrid Learning

**Input:** GNSS data ($x_t$), vehicle interaction graphs ($\tilde{A}$), onboard sensor data. **Output:** Predicted vehicle positions.

1. **Temporal Modeling with LSTM:**
   - Initialize states $(h_0, C_0)$.
   - For each time step $t$, compute: $h_t, C_t = \text{LSTM}(x_t, h_{t-1}, C_{t-1})$
   - Generate temporal embeddings $H_t = [h_1, h_2, \dots, h_T]$.
2. 2. **Spatial Modeling with GCN:**

- Construct graph with adjacency matrix $\tilde{A}$.
- Update node features using: $H^{(k+1)} = \sigma(\widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2}H^{(k)}W^{(k)})$

3. **Feature Fusion with Attention:**
   - Compute attention scores for embeddings: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
   - Fuse temporal and spatial features into a unified embedding.
4. **Federated Training with FedAvg:**
   - Train local models on vehicle-specific data.
   - Aggregate parameters: $w_t = \frac{\sum_{i=1}^{N} n_i w_t^i}{\sum_{i=1}^{N} n_i}$
5. **Prediction:**
   - Use the fused embeddings to predict positions.

### 3.5 Decentralized Communication

Model updates are exchanged between vehicles through direct communication, using protocols like IEEE 802.11p or LTE-V2X. These allow vehicles within a certain range to share data efficiently, even when they are in motion. A mechanism for discovering nearby vehicles dynamically ensures that connections adjust to changing network structures. Gossip protocols distribute updates incrementally, sending information to a small group of nearby vehicles, which then pass it on to others. This creates a chain of communication, reducing the overall burden on the network while keeping the updates consistent across participants.

Privacy is maintained by transmitting only model updates rather than raw data. Differential Privacy adds controlled noise to the shared parameters, making it harder to extract individual vehicle data. Homomorphic Encryption allows computations on encrypted updates, so there is no need to decrypt sensitive information during processing. These methods together create a way for vehicles to collaborate on model training without exposing their private data. This setup supports secure and adaptable communication in constantly changing traffic conditions.

### 3.6 Federated Aggregation

Federated aggregation combines model updates from nodes into a global model without sharing raw data. Weighted averaging is used to account for differences in dataset sizes or data quality across nodes. The global model is updated using the formula:

$$w_t = \frac{\sum_{i=1}^{N} n_i w_t^i}{\sum_{i=1}^{N} n_i}$$

where $w_t^i$ represents the model parameters from node $i$ at time $t$, $n_i$ is the size of the dataset at node $i$, and $w_t$ is the aggregated global model. This ensures nodes with larger or more representative datasets contribute proportionally.

To ensure data integrity during aggregation, secure aggregation methods are employed. Homomorphic encryption allows updates to remain encrypted throughout the computation, enabling nodes to perform aggregation without accessing individual updates. Blockchain-based consensus algorithms can validate and record the updates, creating a tamper-resistant aggregation process.

Dynamic synchronization addresses node connectivity issues. Nodes with intermittent connections can still participate by sharing their updates asynchronously. Missing updates are approximated using prior contributions:

$$\tilde{w}_t^i = \alpha w_t^{i-1} + (1-\alpha) w_t^i$$

where $\tilde{w}_t^i$ represents the approximated update, $w_t^{i-1}$ is the last known contribution, and $\alpha$ is a weighting factor. Outdated updates are excluded using a threshold to maintain accuracy. Synchronization ensures that aggregation continues smoothly, even with delayed or missing inputs.

**Input:** Local models $w_t^i$ and dataset sizes $n_i$ from $N$ nodes. **Output:** Updated global model $w_t$.

1. Initialize the global model $w_t = 0$.
2. For each node $i = 1,2,\ldots,N$: $w_t = w_t + n_i w_t^i$
3. Normalize the weighted sum: $w_t = \frac{w_t}{\sum_{i=1}^{N} n_i}$
4. If secure aggregation is used:
- Encrypt $w_t^i$ using homomorphic encryption.
- Perform aggregation on encrypted updates.
- Decrypt the final result.
5. For missing or delayed updates: $\tilde{w}_t^i = \alpha w_t^{i-1} + (1-\alpha) w_t^i$
6. Replace $w_t^i$ with $\tilde{w}_t^i$ if $w_t^i$ is missing.
7. Exclude outdated updates exceeding a gradient-staleness threshold.
8. Return the aggregated global model $w_t$.

### 3.7 Workflow of the framework

The workflow begins with processed input features derived from GNSS data, onboard sensors, and V2V communication. GNSS provides time-series location data, sensors capture motion-related information like speed and acceleration, and V2V communication delivers spatial relationships, including distances and relative velocities between vehicles. These features are fed into a hybrid machine learning model designed to handle temporal and spatial patterns effectively. The temporal component is managed by an LSTM network, which processes sequential GNSS and sensor data. The LSTM uses its internal states to capture dependencies across time, producing embeddings that represent temporal movement trends. Simultaneously, spatial relationships are modeled using a GCN, which constructs a graph where nodes represent vehicles and edges represent interactions derived from V2V data. The GCN updates the feature representations of nodes by aggregating information from their neighbors in the graph.

The outputs of the LSTM and GCN are fused using a transformer-based attention mechanism. This mechanism assigns importance scores to both temporal and spatial features, ensuring the most relevant aspects of the data are emphasized. The fused representation is passed through a final prediction layer, which outputs the vehicle’s current position and forecasts its trajectory. By combining temporal, spatial, and attention-driven fusion, this workflow produces accurate predictions that account for both historical motion and real-time interactions between vehicles. The model is trained locally on each node using the processed input features, enabling decentralized learning while maintaining data privacy.

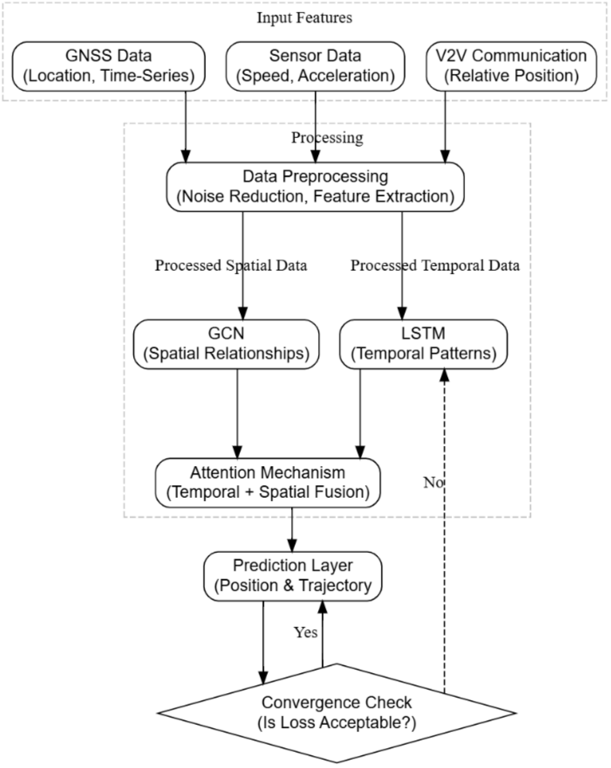

Figure 1: Workflow of the Hybrid Model for Vehicle Localization and Prediction

The workflow diagram shown in figure 1 illustrates the hybrid model’s process for vehicle localization and trajectory prediction. Input data, including GNSS time-series, sensor measurements, and V2V communication, undergoes preprocessing to extract temporal and spatial features. Temporal data is processed by an LSTM to identify time-based patterns, while spatial relationships are modeled using a GCN. These outputs are fused using an attention mechanism to highlight critical features. The fused representation is passed to a prediction layer to estimate vehicle positions and trajectories. A convergence check evaluates whether the model's performance meets predefined criteria. If the loss is unacceptable, the workflow iterates back to refine temporal patterns. The process continues until convergence is achieved, ensuring accuracy in predictions.

## 4 Experimental Study

This experimental study focuses on assessing the performance of the Decentralized Federated Localization Framework (DFLF) in dynamic vehicular networks. The primary objective is to validate the framework’s ability to deliver precise real-time vehicle localization by addressing three critical challenges: mitigating GNSS signal inaccuracies, ensuring data privacy during collaborative model training, and maintaining adaptability in highly dynamic traffic scenarios with changing network topology. Key challenges include GNSS inaccuracies caused by signal obstruction or multipath effects in urban environments, such as tunnels or areas surrounded by tall buildings. Additionally, the need to exchange model updates rather than raw data introduces complexities in preserving data privacy while maintaining efficient communication. The highly dynamic nature of vehicular networks, where vehicles frequently enter and exit the communication range, further complicates accurate localization and model synchronization. To address these challenges, the study employs a simulation-based approach using synthetic GNSS data, onboard sensor measurements, and vehicle-to-vehicle (V2V) communication data. The experimental design leverages a hybrid machine learning model: LSTMs for processing temporal GNSS and sensor data, GCNs for capturing spatial relationships in V2V interactions, and a transformer-based attention mechanism for combining these features. Federated learning techniques, supported by privacy-preserving measures like differential privacy and homomorphic encryption, ensure secure and decentralized model training. The framework's performance is evaluated across different traffic scenarios, including dense urban and highway environments, to measure its accuracy, scalability, and communication efficiency.

### 4.1 Simulation Environment

The simulation setup uses SUMO to replicate diverse traffic scenarios, including dense urban streets and open highways, with realistic vehicle movements, dynamic traffic densities, and frequent changes in vehicular topology. To generate GNSS synthetic data, SUMO is integrated with realistic urban mobility models, simulating GNSS signal inaccuracies such as multipath effects and signal blockages. V2V data, including relative distances, velocities, and positions, is modeled using SUMO's TraCI interface, which facilitates real-time vehicular communication data exchange. IEEE 802.11p, a dedicated short-range communication protocol, facilitates low-latency, high-speed V2V and V2I communication, ensuring uninterrupted data flow even under conditions of high vehicle turnover or network congestion. Privacy is preserved using differential privacy, where Gaussian noise is added to federated model updates. This ensures sensitive information is protected while maintaining the statistical utility of the aggregated models. To address GNSS and sensor data inaccuracies caused by environmental obstructions, Kalman filters are applied. These filters iteratively predict and refine position estimates, smoothing out noise and inconsistencies in the input data.

This simulation environment, combining synthetic GNSS and V2V data sources, robust communication protocols, privacy-preserving methods, and noise reduction techniques, provides a comprehensive platform to evaluate the DFLF framework's performance in achieving accurate, private, and adaptive localization.

### 4.2 Results and Discussion

The experimental study validated the Decentralized Federated Localization Framework (DFLF) as an effective solution for precise real-time vehicle localization in dynamic vehicular networks. The results illustrate its superior performance over contemporary models ST-FTL [7] and FedVCP [18] across various metrics, including localization accuracy, privacy preservation, and scalability. The DFLF framework consistently achieved lower RMSE values compared to ST-FTL and FedVCP across different traffic scenarios. In urban canyon environments, where GNSS signals are prone to interference, DFLF recorded an RMSE of 2.3 meters, outperforming ST-FTL, which achieved 3.1 meters, and FedVCP, which showed a higher error of 4.7 meters. On highways with minimal signal disruptions, DFLF further reduced the RMSE to 1.2 meters, a significant improvement over ST-FTL's 1.8 meters and FedVCP 2.5 meters as shown in table 1. These results highlight the framework's ability to adapt to varying conditions by leveraging hybrid data inputs from GNSS, onboard sensors, and V2V communication.

Table 1: Localization Accuracy Across Different Traffic Scenarios

| Scenario | DFLF (RMSE in meters) | ST-FTL (RMSE in meters) | FedVCP (RMSE in meters) |
|---|---|---|---|
| **Urban Canyon** | 2.3 | 3.1 | 4.7 |
| **Highway** | 1.2 | 1.8 | 2.5 |

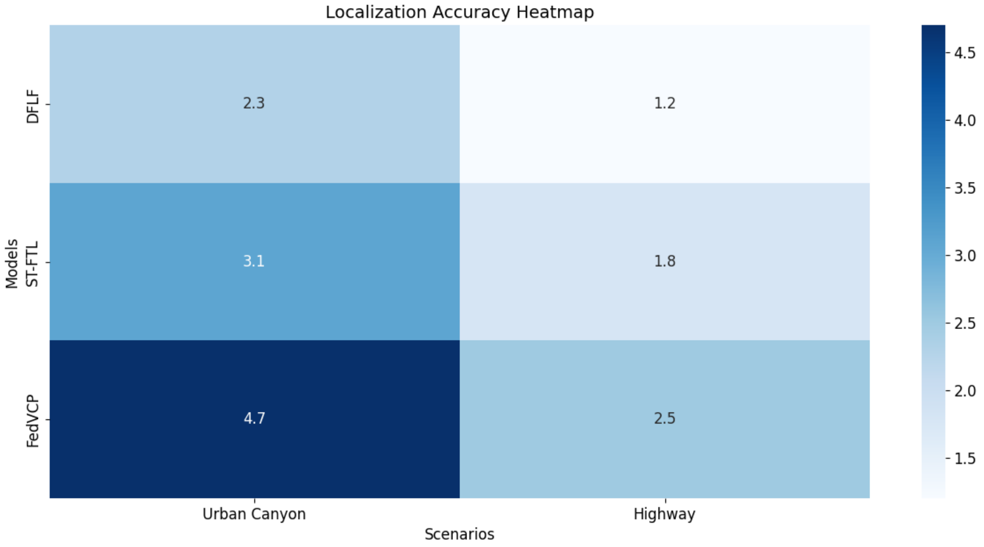

Figure 2: Accuracy Comparison of Models Under Different Scenarios

Localization accuracy was shown in figure 2 achieved through the combined effect of Kalman filtering and the hybrid machine learning model. Kalman filters effectively smoothed noisy GNSS data, reducing errors caused by multipath interference and signal blockages. The integration of temporal and spatial patterns using LSTM and GCN further improved predictions, enabling the framework to maintain consistent accuracy in diverse conditions.

The table 2 study demonstrated that DFLF achieved a strong balance between privacy and performance. When differential privacy was applied, DFLF exhibited an RMSE increase of only 1.5%, a marginal degradation compared to ST-FTL’s 3.8% and FedVCP 7.4%. This minimal loss in accuracy demonstrates the effectiveness of DFLF’s federated learning architecture, which protects sensitive data by sharing only encrypted model updates with added Gaussian noise.

Table 2: Privacy vs. Performance Impact on Localization Accuracy

| Model | RMSE Without Privacy (meters) | RMSE With Privacy (meters) | Accuracy Degradation (%) |
|---|---|---|---|
| **DFLF** | 2.3 | 2.4 | 1.5 |
| **ST-FTL** | 3.1 | 3.3 | 3.8 |
| **FedVCP** | 4.7 | 5.0 | 7.4 |

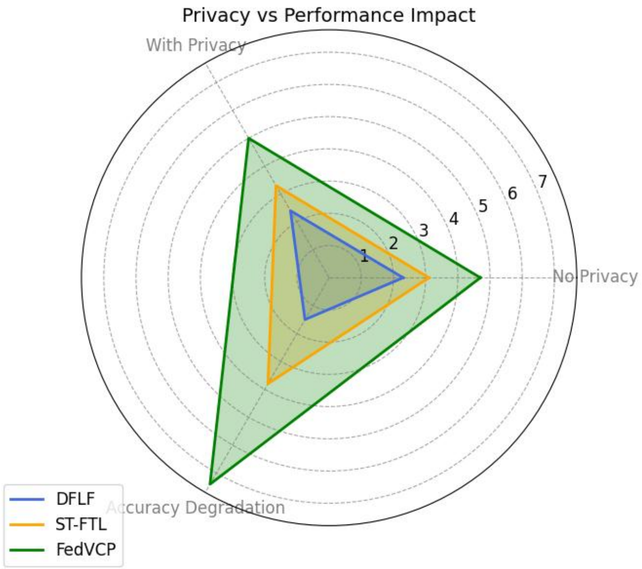

Figure 3: Impact of Privacy Measures on Model Performance

The figure 3 framework's use of differential privacy ensured secure collaboration among vehicles without compromising the statistical utility of the aggregated model. Homomorphic encryption further enhanced the security of updates during federated aggregation, addressing privacy concerns while maintaining competitive performance.

The table 3 and Figure 4 Scalability analysis revealed that DFLF maintained its localization accuracy even as the network size increased. For a network with 100 vehicles, DFLF recorded an RMSE of 2.3 meters, while ST-FTL and FedVCP showed RMSEs of 3.4 meters and 5.1 meters, respectively. With an increase to 500 vehicles, DFLF's RMSE increased slightly to 2.8 meters, compared to 4.6 meters for ST-FTL and 6.8 meters for FedVCP. This result underscores DFLF's efficiency in managing communication and data processing in large-scale networks, attributed to its optimized federated learning mechanism and transformer-based attention for effective feature prioritization.

Table 3: Scalability Analysis of DFLF, ST-FTL, and FedVCP with Increasing Vehicle Density

| **Number of Vehicles** | **DFLF (RMSE in meters)** | ST-FTL **(RMSE in meters)** | FedVCP **(RMSE in meters)** |
|---|---|---|---|
| **100** | 2.3 | 3.4 | 5.1 |
| **500** | 2.8 | 4.6 | 6.8 |

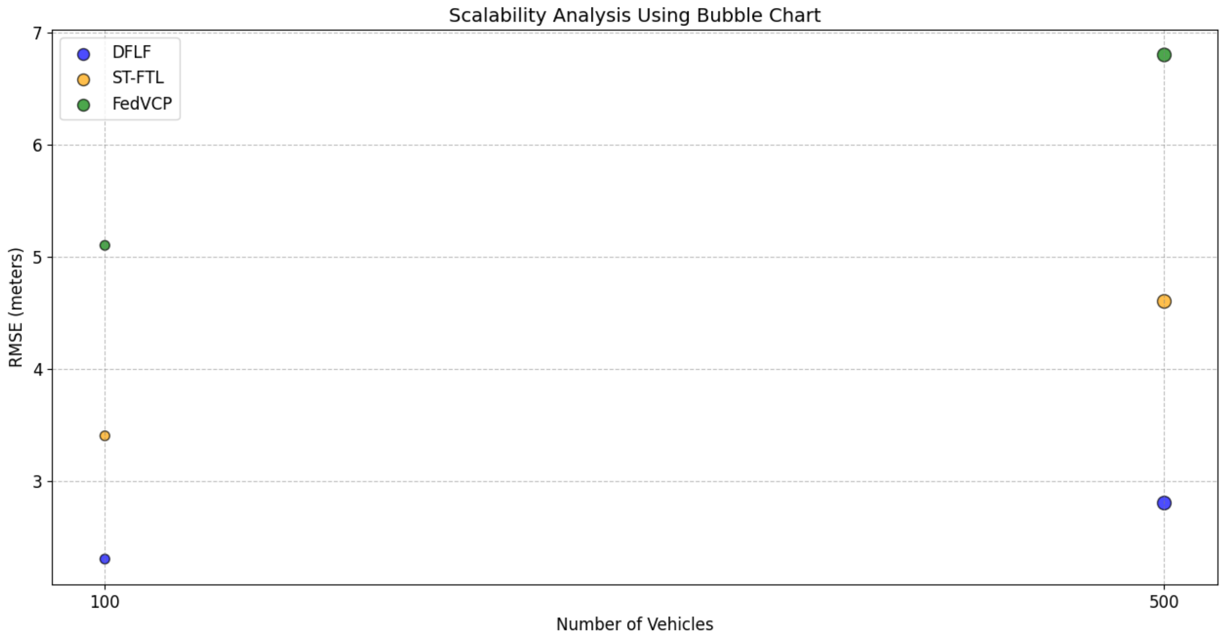

Figure 4: Scalability of Models Across Varying Vehicular Densities

In specific scenarios as shown in Table 4, such as urban canyons and intermittent connectivity, DFLF demonstrated its adaptability and robustness. In urban canyons, the framework leveraged its transformer-based attention mechanism to integrate GNSS, sensor, and V2V data, achieving an RMSE of 2.3 meters, significantly lower than ST-FTL's 3.1 meters and FedVCP 4.7 meters. During intermittent connectivity, DFLF utilized its dynamic synchronization mechanism to compensate for missing updates, maintaining an RMSE of 3.0 meters, outperforming ST-FTL (4.2 meters) and FedVCP (6.3 meters).

Table 4: Case Study Results in Urban Canyon and Intermittent Connectivity Scenarios

| **Scenario** | **DFLF (RMSE in meters)** | ST-FTL **(RMSE in meters)** | **FedVCP (RMSE in meters)** |
|---|---|---|---|
| **Urban Canyon** | 2.3 | 3.1 | 4.7 |
| **Intermittent Connectivity** | 3.0 | 4.2 | 6.3 |

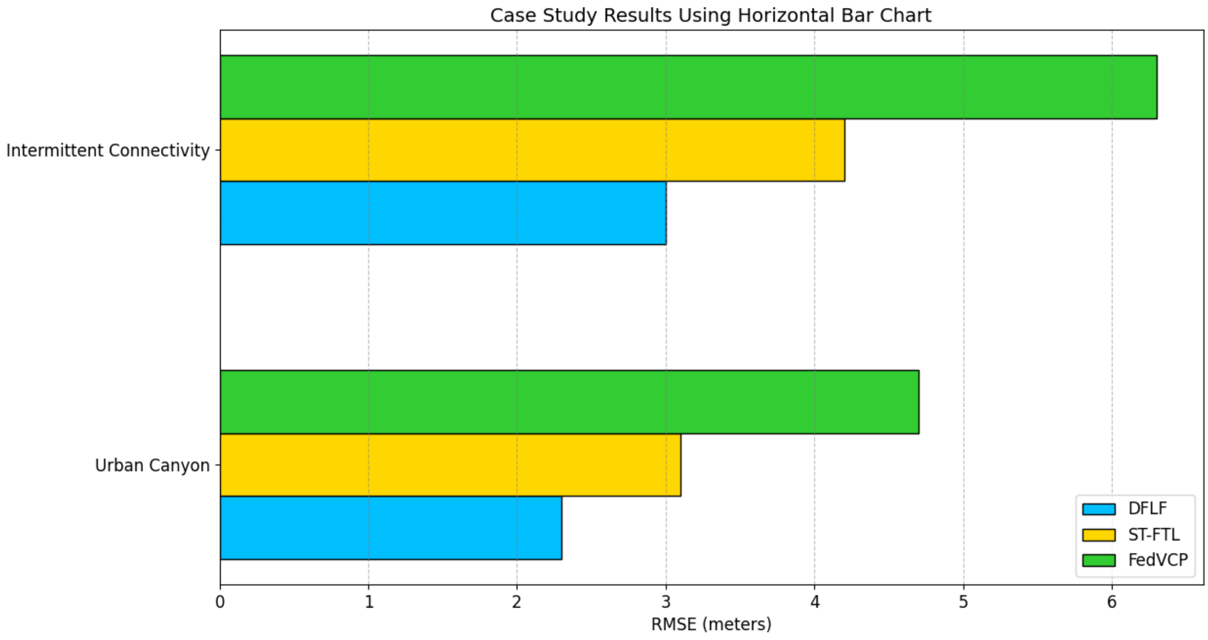

Figure 5: Model Performance in Specific Traffic Scenarios

The figure 5 findings confirm that DFLF successfully addresses the challenges of real-time localization in dynamic vehicular networks. By combining Kalman filtering, LSTM, GCN, and transformer-based attention, the framework effectively integrates temporal and spatial data to produce accurate predictions. The federated learning approach, supported by differential privacy and homomorphic encryption, ensures secure and decentralized model training without significant loss of performance.

Compared to ST-FTL and FedVCP, DFLF demonstrated superior accuracy, scalability, and privacy preservation. ST-FTL, despite having better performance than FedVCP, fell short of DFLF in handling high-density traffic scenarios and balancing privacy and accuracy. The advanced feature fusion in DFLF, driven by the attention mechanism, provided a notable advantage in challenging conditions such as urban canyons and intermittent connectivity.

The study concludes that DFLF offers a robust solution for vehicular localization, addressing the key limitations of existing models. Future research could explore adaptive attention mechanisms and the integration of alternative communication protocols to further enhance the framework's adaptability and efficiency.

## 5 Conclusion

The study focused on addressing real-time vehicle positioning challenges in dynamic environments by developing a Decentralized Federated Localization Framework (DFLF). The framework used data from GNSS, onboard sensors, and V2V communication to improve position estimation. It incorporated machine learning techniques like LSTMs and GCNs, combined with attention mechanisms, to analyze temporal and spatial data. Results showed reduced localization errors, with RMSE values of 2.3 meters in urban areas and 1.2 meters on highways. Scalability was maintained with vehicle networks ranging from 100 to 500 nodes. The research provided a method to handle GNSS limitations and frequent topology changes in vehicular networks. Privacy was preserved by exchanging encrypted model updates, ensuring sensitive data was not shared directly. The approach showed consistent performance under dynamic conditions, such as urban canyons and intermittent connectivity. The study relied on simulations, which might not fully represent real-world situations. Testing with live vehicular data and different communication protocols could provide additional validation. Exploring more adaptive models and integration with other sensor types may improve future designs. This work addressed gaps in decentralized vehicle localization and proposed a framework for handling dynamic traffic scenarios. It outlined a structured approach to achieving precise positioning while managing privacy concerns and communication challenges in vehicular networks.

## References

[1] Barbieri, Luca, Mattia Brambilla, and Monica Nicoli. "A Compressed Decentralized Federated Learning Frame Work for Enhanced Environmental Awareness in V2V Networks." In 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pp. 374-378. IEEE, 2024.

[2] Su, Dongyuan, Yipeng Zhou, and Laizhong Cui. "Boost decentralized federated learning in vehicular networks by diversifying data sources." In 2022 IEEE 30th International Conference on Network Protocols (ICNP), pp. 1-11. IEEE, 2022.

[3] Héry, Elwan, Philippe Xu, and Philippe Bonnifait. "Consistent decentralized cooperative localization for autonomous vehicles using LiDAR, GNSS, and HD maps." Journal of Field Robotics 38, no. 4 (2021): 552-571.

[4] Ashour, Mahmoud, Ahmed Hamdi Sakr, and Rui Guo. "Distributed cooperative localization framework via consensus optimization." In 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), pp. 1-7. IEEE, 2020.

[5] Wang, D., K. O'Keefe, and M. G. Petovello. "Decentralized cooperative positioning for vehicle-to-vehicle (V2V) application using GPS integrated with UWB range." In Proceedings of the ION 2013 Pacific PNT Meeting, pp. 793-803. 2013.

[6] Ciftler, Bekir Sait, Abdullatif Albaseer, Noureddine Lasla, and Mohamed Abdallah. "Federated learning for localization: A privacy-preserving crowdsourcing method." arXiv preprint arXiv:2001.01911 (2020).

[7] Zhou, Xiaokang, Qiuyue Yang, Qiang Liu, Wei Liang, Kevin Wang, Zhi Liu, Jianhua Ma, and Qun Jin. "Spatial–Temporal Federated Transfer Learning with multi-sensor data fusion for cooperative positioning." Information Fusion 105 (2024): 102182.

[8] Parker, Ryan, and Shahrokh Valaee. "Vehicle localization in vehicular networks." In IEEE Vehicular Technology Conference, pp. 1-5. IEEE, 2006.

[9] Pollicino, Francesco, Samih Eisa, Pedro Rosa, Miguel L. Pardal, and Mirco Marchetti. "Decentralized position detection for moving vehicles." In 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), pp. 1-6. IEEE, 2023.

[10] Leung, Keith YK, Timothy D. Barfoot, and Hugh HT Liu. "Decentralized localization for dynamic and sparse robot networks." In 2009 IEEE International Conference on Robotics and Automation, pp. 3135-3141. IEEE, 2009.

[11] Liu, Xiaoyan, Zehui Dong, Zhiwei Xu, Siyuan Liu, and Jie Tian. "Enhanced decentralized federated learning based on consensus in connected vehicles." arXiv preprint arXiv:2209.10722 (2022).

[12] Movahedian, Mahtab, Mahdi Dolati, and Majid Ghaderi. "Adaptive Model Aggregation for Decentralized Federated Learning in Vehicular Networks." In 2023 19th International Conference on Network and Service Management (CNSM), pp. 1-9. IEEE, 2023.

[13] Fu, Huiyuan, Jun Guan, Feng Jing, Chuanming Wang, and Huadong Ma. "A real-time multi-vehicle tracking framework in intelligent vehicular networks." China Communications 18, no. 6 (2021): 89-99.

[14] Santos, Frances A., Ademar T. Akabane, Roberto S. Yokoyama, Antonio AF Loureiro, and Leandro A. Villas. "A roadside unit-based localization scheme to improve positioning for vehicular networks." In 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1-5. IEEE, 2016.

[15] Lee, Soomok, Jung-Hoon Kim, Jung-Woo Kim, GyuMin Oh, and Seung-Woo Seo. "Decentralized Localization Framework using Heterogeneous Map-matchings." In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2183-2189. IEEE, 2018.

[16] Al-Hattab, Mohammad, Maen Takruri, Hussain Attia, and Huthaifa Al-Omari. "Decentralized localization in wireless sensor networks." In 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), pp. 1-5. IEEE, 2017.

[17] Mohammadabadi, Peyman Hadi, and Shahrokh Valaee. "Cooperative node positioning in vehicular networks using inter-node distance measurements." In 2014 IEEE 25th annual international symposium on personal, indoor, and mobile radio communication (PIMRC), pp. 1448-1452. IEEE, 2014.

[18] Kong, Xiangjie, Haoran Gao, Guojiang Shen, Gaohui Duan, and Sajal K. Das. "FedVCP: A federated-learning-based cooperative positioning scheme for social internet of vehicles." IEEE Transactions on Computational Social Systems 9, no. 1 (2021): 197-206.