Scalable Federated Clustering and Pattern Recognition for Multi-Modal Traffic Data in Smart Cities

A Project Report
Submitted in the partial fulfilment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE ENGINNERING

Bejagama Ambica - 2210030039

Sindhuja Reddy - 2210030089

Nagalakunta Sonalika - 2210030127

Medhansh Varma - 2210030119

Under the Esteemed Guidance of

Mr Chiranjeevi Nuthalapati
Assistant Professor
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation

K L (Deemed to be) University
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Declaration

The Project Report entitled **Scalable Federated Clustering and Pattern Recognition for Multi-Modal Traffic Data in Smart Cities**

Is a record of Bonafide work of
Bejagama Ambica - 2210030039
Sindhuja Reddy - 2210030089
Nagalakunta Sonalika - 2210030127
Medhansh Varma – 2210030119

team members submitted in partial fulfilment for the award of B. Tech in Computer Engineering to the K L University, Aziznagar. The results embodied in this report have not been copied from any other departments/University/Institute.

ACKNOWLEDGEMENT

I would like to begin by expressing my sincere gratitude to **Dr. A. Rama Krishna**, Principal of KL University Aziznagar, for providing an excellent academic environment, continuous encouragement, and institutional support that greatly contributed to the successful completion of this project.

I extend my heartfelt thanks to the Head of the Department of Computer Science and Engineering, **Dr. Arpita Gupta**, for her guidance, support, and for fostering a research-driven atmosphere that enabled me to pursue this work effectively.

I am deeply grateful to my project guide, **Mr. Chiranjeevi Nuthalapati**, for his continuous mentorship, invaluable suggestions, and constant encouragement throughout the course of this project. His expertise, timely feedback, and support played a crucial role in shaping the direction and quality of this work.

I would also like to thank the Department of Computer Science and Engineering, KL University, for providing the essential resources, infrastructure, and academic support required for the successful execution of this project.

My heartfelt appreciation goes to my faculty members, peers, and friends for their motivation, constructive suggestions, and assistance during various stages of the project.

Table of Contents

Abstract

Smart cities rely increasingly on dense, heterogeneous sensor infrastructures—such as LiDAR scanners, multi-camera systems, and GNSS/IMU navigation units—to monitor roadway activity, detect congestion, and support autonomous mobility. These sensors generate large volumes of high-frequency data, much of which is unstructured and computationally expensive to process. Traditional centralized traffic analytics pipelines require continuous uploading of raw sensor streams to the cloud, resulting in substantial bandwidth consumption, latency issues, and significant privacy risks. A single LiDAR frame may contain more than 100,000 three-dimensional points, and transmitting such raw data at scale is neither economical nor practical for real-time ITS operations. Furthermore, centralized storage of navigation traces and 3D spatial data may expose sensitive environmental, positional, or behavioural information, making privacy a critical concern.

To address these limitations, this project proposes a federated clustering framework that decentralizes computation by enabling edge nodes—such as roadside units or connected vehicles—to locally process multimodal sensor data. Instead of forwarding complete LiDAR scans or high-dimensional feature vectors, each node extracts compact statistical descriptors, including LiDAR-based geometric features (mean range, variance, point density, height dispersion) and OXTS-derived motion features (velocity, acceleration, orientation). Unsupervised clustering algorithms such as K-Means and DBSCAN are applied locally to characterize traffic states, and only the resulting cluster centroids and metadata are transmitted to a central server. This greatly reduces communication overhead while preserving user privacy and computational efficiency.

A representative subset of the KITTI Raw Dataset is used to demonstrate the proposed system. A simulated multi-node federated environment reflects real-world distributed conditions where each node observes only local traffic dynamics. The central server performs weighted centroid fusion and meta-clustering to assemble a global picture of city-scale traffic patterns. Experimental evaluations include cluster quality metrics such as Silhouette Score and Davies–Bouldin Index, as well as bandwidth analysis comparing raw-data transmission with centroid-based communication. Results show that even with highly compressed feature representations, the system effectively captures key traffic regimes—such as dense congestion, medium flow, and sparse open-road conditions—and reconstructs stable global patterns through federated aggregation.

Beyond empirical validation, this work highlights broader considerations for deploying federated analytics in smart city infrastructures, including privacy preservation, system robustness under heterogeneous data distributions, and potential extensions involving deep feature extraction, hierarchical federated architectures, and integration with real-time traffic management systems. Overall, the project demonstrates that federated unsupervised learning is a scalable, efficient, and privacy-preserving alternative to centralized traffic analytics, enabling smart cities to obtain continuous situational awareness without compromising ethical, communication, or computational constraints.

Keywords: Federated Clustering, LiDAR, KITTI, Traffic Density Estimation, Edge Computing, Privacy Preservation, Unsupervised Learning, OXTS Navigation, Smart Cities.

1. Introduction

1.1 Background

Rapid urbanization and increasing vehicle ownership have made traffic management one of the most pressing operational challenges for modern cities. Intelligent Transportation Systems (ITS) aim to improve mobility, reduce congestion, and increase road safety by combining sensing technologies, data-driven analytics, and automated control. Common sensing modalities used in ITS deployments include camera imagery, Light Detection and Ranging (LiDAR) point clouds, radar, and GNSS/IMU units. Each modality provides complementary information: cameras deliver high-resolution visual context, LiDAR provides accurate 3-dimensional spatial structure and object range, radar offers robust velocity estimation under adverse weather, and GNSS/IMU (e.g., OXTS) provide ego-vehicle pose, velocity and inertial measurements.

While modern sensors generate rich information, they also produce massive volumes of raw data. For example, a single high-resolution LiDAR frame can contain hundreds of thousands of 3D points; high-resolution camera images (30+ fps) add still more. Transmitting, storing, and centrally processing all this raw data at city scale is costly, introduces latency that is incompatible with some real-time applications, and raises privacy concerns when raw imagery or trajectories could be reconstructed. These practical constraints motivate decentralized analytics approaches in which data is processed near the sensors (at the edge) and only compact summaries, model updates or aggregates are transmitted to a central server for global insight.

Federated learning and federated analytics are paradigms that allow multiple data holders to collaboratively learn models without sharing raw data. While most federated-learning work focuses on supervised model training (e.g., federated averaging of neural network weights), a less-explored but highly practical avenue is federated unsupervised learning, particularly federated clustering. In this paradigm, edge nodes compute local cluster summaries (centroids, counts, variances) and share only these compact objects. The aggregator merges these summaries into global clusters that reflect city-scale traffic modes (e.g., free-flow, congested, stop-and-go). This approach reduces communication cost, improves privacy, and scales to large sensor fleets.

1.2 Motivation

Key motivations for this project are:

Bandwidth limits and latency requirements. Real-time traffic control needs fast, frequent updates. Sending full point clouds and images over cellular or constrained municipal networks is often infeasible.

Privacy and data minimization. Raw video and precise vehicle trajectories may reveal personally identifiable information (license plates, behavioral traces). Exchanging compressed statistical summaries reduces the risk of sensitive disclosure.

Heterogeneity and scalability. Sensors differ by type, sampling rate and quality. A federated approach naturally accommodates different local processing capabilities and lets each node tailor preprocessing while contributing to a global view.

Interpretability and operational value. Compact cluster summaries (centroids and labels such as 'dense/medium/sparse') are easier for traffic operators to inspect, visualize, and use for control decisions.

1.3 Objectives

This project aims to design, implement, evaluate and document a federated clustering pipeline for multi-modal traffic data with the following objectives:

O1 — Compact, informative features: Define a small set of per-frame LiDAR and OXTS features (e.g., mean range, range std, forward speed) that capture traffic state.

O2 — Edge clustering protocol: Implement efficient local clustering (K-Means and alternatives) suitable for devices with limited compute and memory.

O3 — Aggregation mechanics: Design aggregation algorithms (concatenate+recluster, weighted averaging, matching) that fuse node centroids into meaningful global clusters.

O4 — Empirical validation: Use the KITTI raw subset + simulated federated nodes to measure cluster quality (Silhouette, Davies–Bouldin), communication savings, and robustness to heterogeneity.

O5 — Privacy & deployment considerations: Propose mechanisms (differential privacy, secure aggregation) and a practical deployment plan.

1.4 Contributions

The primary contributions of this work are:

A practical federated clustering pipeline that extracts compact LiDAR/OXTS feature vectors, runs local clustering, and aggregates centroids with weighted fusion strategies.

A working prototype implementation (Colab notebook) that demonstrates data ingestion from KITTI raw files, point-cloud feature computation, simulated edge splits, local clustering, and global aggregation with visualizations (PCA plots, density scatter, 3D LiDAR renders).

An empirical evaluation that quantifies clustering quality and communication savings for several federated settings, with sensitivity studies over node counts and noise.

A deployment and privacy blueprint describing how to move from prototype to pilot deployment, including choices for edge hardware, communication protocols, and privacy mechanisms.

2. Literature Review

2.1 Federated Learning — overview and key ideas

Federated Learning (FL) is a distributed learning paradigm in which model training is performed across many clients (devices or nodes) that keep their raw data locally and only exchange model updates or compact summaries with a central server. The canonical algorithm, Federated Averaging (FedAvg), aggregates locally computed weight updates to form a global model without centralizing private data; FL has since broadened into a system-level field that addresses issues of communication efficiency, client heterogeneity, privacy guarantees, and system robustness. Early and highly cited surveys lay out the system components of FL (client selection, communication protocols, aggregation rules, and privacy mechanisms) and emphasize trade-offs between model utility, communication cost, and privacy risk.

Key technical motifs in FL include compression and quantization of updates to save bandwidth, asynchronous or partial participation strategies to handle intermittent clients, personalization (client-specific model adaptation), and privacy-enhancing techniques such as differential privacy (DP) and secure aggregation. These motifs form a toolkit that can be adapted to unsupervised tasks (e.g., clustering) and domain-specific deployments (e.g., ITS).

2.2 Unsupervised federated learning and federated clustering

While most FL research concentrates on supervised model training (classification/regression), unsupervised federated learning — and specifically federated clustering — has been gaining attention because many real-world scenarios lack labeled data or require distributional analysis rather than predictive modeling. Federated clustering typically follows one of two paradigms: (1) centroid/summary exchange, where clients compute local cluster centers (centroids), counts and variance summaries and transmit these compact objects to the server for global aggregation; or (2) model-parameter sharing of unsupervised representation learners (e.g., autoencoders) that produce shared latent features for downstream clustering. The centroid-exchange approach is particularly attractive for resource-constrained edge nodes because centroids are small, interpretable, and cheap to communicate. Recent algorithmic work has proposed principled ways to reconcile local clusterings into a global clustering, including matching local centers to global centers, weighting by cluster sizes, and one-shot recovery algorithms that adaptively refine local centroids before aggregation. Notably, new methods such as Federated Centroid Aggregation (FeCA) and similar one-shot aggregation schemes aim to recover global k-means solutions by refining and aggregating local centers with theoretical and empirical support. Other unsupervised FL approaches (e.g., Orchestra) explore globally consistent clustering by alternating local updates and constrained global clustering steps to create discriminative partitions without labels. These methods highlight that unsupervised federated learning is possible beyond naive averaging but requires careful algorithmic design to handle non-IID client data and label/permutation alignment issues.

2.3 Clustering methods used for LiDAR and traffic analytics

At the sensor/feature level, classical clustering algorithms remain heavily used for LiDAR and roadside perception tasks where labels are scarce and object segmentation or density estimation is required. K-Means is a widely used baseline for partitioning compact feature vectors (e.g., mean range, height statistics), while DBSCAN and other density-based algorithms are favored for raw point-cloud segmentation and detection because they can discover arbitrarily-shaped clusters and ignore noise/outliers. Two-stage pipelines (pre-filtering/ground removal → DBSCAN/region-growing → postmerge) are common in infrastructure LiDAR applications for traffic monitoring due to their robustness and low latency. Systematic reviews of clustering for sensor data emphasize adapting parameters (DBSCAN eps/minPts) dynamically to account for range-dependent point sparsity in LiDAR and propose occupancy-grid or histogram features to compactly represent point clouds when exact segmentation is unnecessary.

For traffic pattern analysis, compact features such as average LiDAR range, range variance, per-frame point counts (density proxies), and vehicle speed from OXTS/GNSS sensors have been shown to capture traffic regimes (free-flow vs congested). Combining geometric LiDAR summaries with kinematic features yields robust clustering in many ITS studies; when higher fidelity is required, researchers augment these summaries with object counts from detectors or use learned point-cloud embeddings (PointNet variants) before clustering.

2.4 Federated learning in Intelligent Transportation Systems (ITS)

The application of FL to ITS is an active and rapidly evolving area. ITS deployments benefit from FL's privacy-preserving and communication-efficient properties because vehicular and roadside sensors often capture sensitive location and movement information and operate over constrained networks. Surveys and domain-specific reviews (spanning 2020–2024) discuss how FL has been applied to tasks in ITS such as traffic flow prediction, anomaly/incident detection, perception model sharing among vehicles, and cooperative perception for connected and autonomous vehicles (CAVs). These works identify recurrent challenges: high client heterogeneity (different sensors, sampling rates), stringent latency requirements for some control applications, potential model poisoning by compromised clients, and the difficulty of measuring privacy risk in mobility data. Proposed solutions in the ITS literature include hierarchical federated architectures (regional aggregators), compact feature extraction at the edge, and combinations of DP with secure aggregation to balance privacy and utility.

Applied examples demonstrate FL for perception fusion (sharing learned feature extractors or object detection models across vehicles), and for predictive tasks (traffic speed/flow forecasting). For unsupervised analytics like clustering or anomaly detection, the literature is newer but suggests the centroid-exchange paradigm is especially appropriate for roadside LiDAR/occupancy monitoring because it minimizes bandwidth and preserves interpretability for traffic operators.

2.5 Challenges, gaps, and research directions

The literature surfaces several important challenges that directly motivate the system design choices in this project:

Non-IID and heterogeneous data: Sensor characteristics, deployment geometry, and traffic patterns vary across nodes; this heterogeneity can distort naive aggregation and requires weighted fusion, robust matching of centers, or cluster-aware aggregation strategies. Recent cluster-centric FL research explicitly targets these non-IID effects.

Privacy vs utility tradeoffs: Differential privacy and noise injection can protect client summaries but may destroy subtle distributional signals important for clustering; calibrating DP noise for centroid summaries is an active research area.

Communication & compute constraints: Edge devices may have limited memory/CPU and unreliable connectivity, motivating mini-batch local clustering, one-shot aggregation, and compressed centroid encodings. Practical ITS papers recommend lightweight feature extraction on-device followed by small update exchanges.

Robustness & adversarial attacks: Poisoning (malicious centroid updates) and faulty sensors can degrade global clusters; defenses include anomaly detection on updates, robust aggregation rules, and authenticated client updates. These topics appear across recent FL surveys and ITS-focused reviews.

To summarize, the literature indicates that federated clustering by centroid/summary exchange is a promising and practical approach for ITS scenarios where raw sensor transmission is infeasible or undesirable. Recent algorithmic advances (FeCA, one-shot centroid aggregation, Orchestra-style global constraints) provide methods to align and combine local cluster outputs into coherent global clusters, while domain studies on LiDAR clustering and ITS deployments provide feature engineering and system-level guidance. This project builds on those threads by: (a) using compact LiDAR/OXTS summaries that are recommended by ITS literature, (b) implementing simulated federated centroid exchange and weighted aggregation inspired by federated clustering research, and (c) evaluating the pipeline on KITTI-derived data to quantify clustering quality, communication cost, and robustness — thereby addressing an operationally relevant gap between algorithmic FL research and ITS field deployment work.

2.5 Summary Of Papers

Study	Contribution	Key Insights
Yan et al. (2023, IEEE T-ITS)	Privacy-aware FL for connected BEVs energy consumption	Showed FL can preserve privacy while enabling accurate energy modeling in ITS.
Wang et al. (2023, Elsevier)	Spatio-temporal federated transfer learning with multi-sensor data	Combined FL with transfer learning for richer ITS models.
ACM (2024)	Federated learning for transportation applications	Introduced scalable FL frameworks addressing efficiency bottlenecks.
Li et al. (2023, PSU)	Multimodal FL survey	Comprehensive survey on multimodal FL, highlighting sensor fusion challenges.
MDPI (2024, Information)	FL in smart environments	Case studies in IoT and smart cities; energy-efficient deployment.
IEEE TSUSC (2024)	Privacy-preserving FL for ITS	Proposed lightweight and secure FL framework for real-time applications.
MDPI (2024, ISPRS IJGI)	Federated geospatial sensor fusion	Showed improvements in traffic flow modelling using multi-sensor FL.
MDPI (2024, Electronics)	Edge-oriented FL for ITS electronics	Lightweight FL for vehicle electronics and smart mobility.
Springer (2024, Neural Computing & Applications)	Federated deep learning for traffic recognition	Proposed clustering + recognition models on multimodal traffic data.
MDPI (2022, Sustainability)	Sustainable FL for smart cities	Focused on deployment strategies under energy & urban sustainability constraints.
MDPI (2023, Sustainability)	Federated clustering in ITS	Pioneered FL-based clustering methods for ITS datasets.
Hugging Face (2023, arXiv:2312.07371)	Towards federated multimodal learning for ITS	Addressed autonomous driving with multimodal fusion under FL.

Table 2.1 summarizes recent (2022–2025) federated learning studies in ITS and smart cities, with emphasis on contributions and insights across privacy, multimodal fusion, scalability, and sustainability.

2.6 Research Gaps

Research Gap	Representative Studies	Limitations in Current Work	Opportunities / Future Directions
Handling non-IID and heterogeneous multimodal ITS data	Yan et al. (2023); Li et al. (2023); Wang et al. (2023)	Most methods assume IID or homogeneous data; poor generalization across urban vs. rural contexts; limited multimodal integration.	Develop robust aggregation methods for non-IID multimodal data (LiDAR, GPS, video); adaptive personalization for diverse traffic patterns.
Scalable federated clustering for massive streaming datasets	Springer (2024); MDPI IJGI (2023)	Traditional clustering (K-means, spectral) not scalable for real-time streaming; mostly evaluated on small offline datasets.	Design incremental federated clustering algorithms with online updates; reduce computational cost for city-scale traffic networks.
Real-time model convergence under latency constraints	ACM (2024); IEEE TSUSC (2024)	Multiple communication rounds cause delays; lack of bandwidth-efficient aggregation; limited fault tolerance in mobile edge networks.	Develop communication-efficient FL protocols; asynchronous updates; lightweight edge models for ITS real-time applications.
Trade-off between privacy and model accuracy	Alshamrani et al. (2025); IEEE TSUSC (2024)	Noise injection/differential privacy reduces accuracy; most works don’t jointly optimize privacy, accuracy, and latency.	Create adaptive privacy mechanisms that balance security and utility based on application sensitivity (e.g., personal vs. aggregated data).
Sustainable architectures for smart cities	MDPI (2022, 2023, 2024)	FL requires heavy computation and frequent communication; energy impact rarely measured; not aligned with green AI goals.	Propose energy-efficient FL frameworks; optimize edge–cloud workload distribution; support carbon-neutral smart city deployments.

Table 2.2 Summary of key research gaps, limitations, and future opportunities in federated learning for ITS and smart cities, covering challenges in multimodal data handling, scalability, privacy–accuracy trade-offs, real-time convergence, and sustainable model deployment.

3. Problem Definition

The increasing complexity of urban transportation systems, combined with the rapid adoption of high-resolution sensing modalities such as LiDAR scanners, multi-camera vision units, GNSS/IMU navigation sensors, and vehicular telematics, has fundamentally changed the landscape of modern traffic analysis. Traditional centralized approaches—where all raw sensor data is transmitted to a single processing hub or cloud platform—are gradually becoming impractical for large-scale deployments. This is due to the unprecedented growth in data volume, the stringent requirements of real-time applications, and the rise of data privacy regulations that restrict the sharing of sensitive information. In such environments, cities are increasingly populated with connected vehicles, autonomous driving platforms, smart intersections, roadway instrumentation and IoT-enabled devices, all generating streams of data so dense that previous data-management assumptions no longer hold.

The fundamental problem arises from the fact that raw sensor data is extraordinarily large, multi-modal, and high-frequency. A single LiDAR sensor, depending on its configuration, can generate hundreds of thousands of spatial measurements per frame. At even moderate frame rates—ten or twenty frames per second—this amounts to billions of points per hour from a single device. If a city deploys hundreds or thousands of sensors of this kind, the cumulative bandwidth required to transmit such data to a central server becomes nearly impossible to sustain. Camera sensors contribute even more densely: high-definition video streams, often recorded in RGB at 30 frames per second, result in gigabytes of visual data per hour from each unit. When these sensors are coupled with navigation packets such as OXTS/GNSS readings, which contain granular velocity, acceleration, orientation, and trajectory data, the scale of information increases further. Centralized processing architectures are not suited to manage this exponential increase in data generation, which is projected to grow at rates beyond what network infrastructure can economically support.

The problem becomes even more acute when considering the operational needs of modern intelligent transportation systems (ITS). Many decision-making processes in traffic control, such as congestion detection, adaptive signal timing, dynamic routing, and emergency response prioritization, require information updates within milliseconds or seconds. Centralizing raw sensor data introduces unavoidable delays: the data must first be captured at the node, then transmitted through the network, buffered at the server, processed by a high-performance computing cluster, and finally transmitted back to field controllers or vehicles. These multi-layered, multi-hop delays are often incompatible with real-time constraints, especially during peak traffic periods or emergency conditions. Thus, latency is not merely an inconvenience—it fundamentally degrades the operational value of the entire system.

Privacy concerns further complicate the centralized processing paradigm. Many advanced sensors, especially cameras and LiDAR scanners, inadvertently capture sensitive details such as faces, physical attributes, vehicle license plates, realistic 3D models of pedestrians, or location-based trajectories that reveal behavioral patterns. Regulations such as GDPR, and emerging data protection frameworks in India and other jurisdictions, impose strict rules on how personal data can be collected, stored, transmitted, and used. Centralizing raw sensor data creates a single point of vulnerability, making the system susceptible to unauthorized access, cyberattacks, or unintended data leakage. These concerns are not hypothetical; numerous documented breaches across industries highlight the risks of aggregating highly sensitive data in centralized repositories. Therefore, a technical approach that minimizes the exposure of raw data while still enabling meaningful analytics is not only advantageous but crucial.

Beyond privacy and bandwidth concerns, centralized analytics imposes significant computational challenges. Storing and processing terabytes of LiDAR and video data requires large-scale GPU clusters, petabyte-level storage systems, and expensive data infrastructure. Not only is this financially infeasible for many municipalities, but the processing time required to analyze such massive datasets is often longer than the temporal relevance of the data itself, especially for tasks like real-time traffic monitoring or anomaly detection. Even if scaling infrastructure were financially feasible, the environmental and energy costs of operating large data centers conflict with sustainability goals increasingly adopted by smart city initiatives.

These complex challenges—bandwidth limitations, latency constraints, privacy regulations, and computational overhead—collectively form the core motivation for decentralized and federated approaches. Federated learning, originally conceived to solve privacy constraints in mobile-device environments, provides a new design philosophy for intelligent systems: keep raw data on the device where it is generated, perform local computation, and transmit only compact, non-sensitive model updates or mathematical summaries to a central server. The federated model aligns well with the natural distributed structure of traffic sensing networks. Instead of treating sensors as dumb data collectors, federated approaches empower them to operate as computationally intelligent agents capable of extracting features, performing clustering, and contributing their insights without exposing sensitive or bulky raw data.

Applying federated techniques to unsupervised learning—specifically clustering—brings its own set of complexities. In traditional centralized clustering, a large dataset is pooled together and algorithms such as K-Means, DBSCAN, or hierarchical clustering identify groups or patterns based on global statistical relationships. In federated clustering, however, the dataset is not pooled; each node analyzes only its local subset, which may be biased, incomplete, or non-representative of the city-wide distribution. For example, sensors positioned near intersections may see frequent stop-and-go traffic, while sensors on freeways may observe mostly high-speed, low-density motion. If each node performs clustering independently, their cluster centers will naturally represent different patterns. The challenge is to reconcile these local models—often with different numbers of clusters, different variances, and different interpretations—into a single global model that meaningfully reflects the broader traffic ecosystem.

The core problem of this project can therefore be articulated as follows: How can a large-scale, multi-modal traffic monitoring network generate accurate, city-wide traffic patterns and cluster models without centralizing raw sensor data, while maintaining accuracy, robustness, privacy, and efficiency? This problem requires addressing several intertwined sub-problems. First, how do we design feature extraction functions that reduce high-dimensional LiDAR and OXTS data into compact representations that still reflect key traffic dynamics? Second, how do we design clustering algorithms that operate efficiently on constrained edge devices while maintaining stability despite noisy or incomplete data? Third, how do we aggregate heterogeneous local cluster outputs into a global clustering structure that approximates the solution that would have been achieved if all data were centralized? Fourth, how do we ensure that the system is resilient to real-world issues such as sensor failures, network outages, asynchronous updates, and node heterogeneity? Finally, how do we reason about privacy in a federated clustering context, where even aggregated centroids may leak statistical information if not properly protected?

A formal description of the problem begins with considering a distributed network of sensing nodes deployed across an urban region. Each node continually collects LiDAR point clouds, camera images, or navigation data. The raw data at each node is inherently continuous, voluminous, and multi-modal. The system must transform this data into low-dimensional feature vectors that capture essential traffic characteristics such as object proximity, density, motion, and variability. Once features are extracted, each node must perform local clustering on its own data stream. These local clusters represent the patterns observed at that particular location—for example, periodic congestion, vehicle platoon formation, or frequent pedestrian crossings. The output of the local clustering process consists of cluster centres, cluster sizes, cluster variances, and possibly additional statistical descriptors.

Instead of transmitting raw data, each node must transmit only these compact summaries to a central aggregator. The aggregator must then merge the cluster summaries from all nodes into a coherent global structure. This step is non-trivial because local clusters from different nodes may not correspond one-to-one; some clusters may represent similar traffic patterns even though they are generated from different environments, while others may be unique to specific locations. The aggregator must therefore employ robust strategies for aligning, merging, and weighting cluster centres so that the resulting global model reflects a statistically meaningful view of the entire network.

Beyond this, the problem incorporates additional constraints. The system must preserve the privacy of individuals, preventing reconstruction of identifiable trajectories or scenes. It must remain efficient even when deployed at scale. It must tolerate noise, abrupt changes in traffic behaviour, and variations in sensor calibration. More importantly, it must maintain correctness over time even as traffic conditions evolve due to weather changes, special events, or road incidents.

The central challenge can thus be summarized not as the development of a single model or algorithm, but as the design of an end-to-end system that addresses sensing, feature extraction, clustering, communication, aggregation, privacy, and interpretability within an integrated federated framework.

3.1 Objectives of the Project

The primary objective of this project is to design and develop a comprehensive, scalable, and privacy-preserving federated clustering framework capable of analysing large-scale traffic data gathered from multi-modal sensors in smart city environments. Rather than simply producing a technical prototype, the goal is to create an integrated system that addresses the pressing challenges of data volume, communication limitations, privacy constraints, and real-time computational demands that characterize modern intelligent transportation systems. The project seeks to demonstrate that meaningful global traffic patterns can be inferred from distributed sensing nodes even when raw sensor data never leaves the local devices. This objective reflects a paradigm shift from centralized big-data analytics toward decentralized intelligence that treats edge devices not merely as data collectors but as active participants in the computational process.

A central aim of the project is to transform the vast and complex data collected by LiDAR scanners, OXTS/GNSS units, and other vehicular or roadside sensors into compact but informative representations that can drive unsupervised learning at the edge. Since federated systems cannot depend on raw data transmission, this transformation lies at the heart of the project's objective: the feature extraction pipeline must reduce dimensionality drastically, while retaining the most important structural and semantic properties of the traffic environment. The project therefore focuses on developing robust feature engineering methods capable of summarizing point-cloud geometry, object proximity, height variations, and kinematic attributes such as speed and acceleration. The objective is to create features that not only support effective clustering but are computed efficiently enough to run on modest hardware placed at intersections, traffic lights, or within connected vehicles.

Another significant objective is to explore and refine clustering algorithms suitable for decentralized execution. While conventional clustering techniques like K-Means and DBSCAN work well in centralized environments, their behaviour in distributed, non-IID conditions is far more complex. Each node in a federated system may observe only a partial or biased subset of the global data distribution. For instance, a sensor situated near a highway entrance may observe primarily high-speed, free-flowing traffic, while another placed at a busy intersection may capture sequences of stop-and-go congestion. The objective in this context is to develop a local clustering mechanism that can operate effectively and produce meaningful local cluster representations even when the underlying data distribution is narrow or skewed. Equally important is the requirement that these local cluster outputs must be structured in a manner that allows them to be merged at the server into coherent city-wide patterns.

The project also sets out to design reliable and theoretically grounded aggregation strategies for combining local clustering results into global clusters. Since each node has its own local perspective, the aggregation mechanism must reconcile these differing viewpoints while respecting the statistical significance of each local contribution. One objective is to investigate weighted centroid fusion, centroid matching, and reclustering over the union of local centroids as potential methods for global aggregation. The system must ensure that global clusters represent the holistic traffic conditions across the network rather than being dominated by the data of any single node. This requires careful handling of cluster sizes, variances, dissimilar scales, and noise levels. By accomplishing this, the project aims to show that global clusters produced through federated aggregation approximate—or in some cases even rival—those produced by full data centralization.

A further objective is to demonstrate the practical scalability of the proposed system under realistic constraints. Federated analytics is not merely a mathematical idea but a system-level approach that must operate under the limitations of real-world networks, hardware, and budgets. Therefore, the project aims to evaluate the communication overhead of transmitting cluster summaries compared to raw sensor data and to quantify the efficiency gained by federated processing. Demonstrating that cluster summaries require orders of magnitude fewer bytes than raw frames forms an important part of this objective. In addition, the system must be robust to network interruptions, asynchronous updates, and varying node participation rates. The project therefore seeks to develop a resilient pipeline that remains stable even when nodes drop offline intermittently or contribute updates at irregular intervals.

Privacy preservation forms another cornerstone objective of this work. The project aims to build a framework that aligns with the principles of data minimization and privacy-by-design. Although the system does not transmit raw data, even feature summaries may carry small risks of inference attacks, particularly if adversaries attempt to reconstruct patterns from accumulated updates. The project therefore explores mechanisms that enhance privacy without degrading utility, such as the addition of carefully calibrated noise to centroids, the adoption of secure aggregation techniques, or the alignment of local updates with differential privacy constraints. While the complete implementation of differential privacy is outlined as future work, the project's objective is to structure the architecture so that privacy-enhancing technologies can be fitted naturally at later development stages.

Beyond the computational and system-level objectives, the project also seeks to contribute to the academic understanding of federated unsupervised learning—an area still underexplored when compared to supervised federated learning. Most prior work in FL focuses on classification or regression tasks involving model weights. By contrast, this project aims to expand the literature by demonstrating that clustering, a fundamentally different learning paradigm, can be collaboratively achieved even when data is fragmented across heterogeneous participants. Through experimentation with feature definitions, clustering algorithms, and aggregation strategies, this project aspires to shed light on the unique mathematical and practical considerations that arise when performing unsupervised tasks in federated environments.

Another important objective is to assess the interpretability and operational usefulness of the resulting global clusters for traffic management. While high-dimensional machine learning models often achieve strong performance, they frequently lack intuitive meaning for domain experts such as traffic engineers or city planners. This project aims to design clusters that correspond to understandable traffic regimes—free-flow patterns characterized by long object distances and high speeds, medium-flow conditions with moderate variability, and congested patterns where speeds fall and LiDAR returns become denser. By aligning clusters with real-world semantics, the system becomes more valuable for decision-making and can integrate seamlessly with downstream applications such as adaptive traffic control, incident detection, or congestion monitoring.

In addition to producing a functioning prototype, the project sets out to rigorously evaluate the performance of the federated clustering system in simulated settings. This involves running experiments on real KITTI raw data, splitting features across multiple virtual nodes to mimic federated sensing environments, and comparing local clustering outputs to their aggregated global forms. The evaluation aims to measure not only clustering quality but also communication overhead, resilience to noise, sensitivity to skewed data, and stability across repeated runs. By performing comprehensive testing and visualization—including PCA projections, centroid plots, and heatmaps—the project aims to provide convincing empirical evidence that federated clustering is a viable strategy for multi-modal traffic data analysis.

Finally, the overarching objective of the project is to demonstrate that federated clustering can serve as a foundation for a new generation of intelligent traffic systems where privacy, scalability, robustness, and efficiency coexist without compromise. The work lays the groundwork for future extensions such as hierarchical federated architectures, integration with deep representation learning, and deployment in real smart city infrastructures. By proving the technical feasibility of decentralized clustering in complex sensing environments, the project aims to motivate further research and inspire practical deployments.

4. Dataset Preprocessing

The effectiveness of any intelligent traffic analysis system depends fundamentally on the quality, richness, and representatives of the data on which it is built. In this project, the foundation of the experimental pipeline is the KITTI Raw dataset, one of the most widely used and respected datasets in the fields of autonomous driving, computer vision, sensor fusion, and robotics research. KITTI is particularly valuable because it provides real-world multi-modal sensor data collected from an instrumented vehicle driven through urban, suburban, and highway regions. The dataset captures the complexity, irregularity, and unpredictability that are intrinsic to real-world traffic environments—qualities that synthetic datasets or simulations often fail to replicate. For a system such as the one developed in this project, which aims to explore federated clustering in a realistic and robust manner, the KITTI Raw dataset offers an ideal testbed.

The vehicle used to record the KITTI Raw dataset is equipped with a sophisticated suite of sensors, including a high-resolution Velodyne HDL-64E LiDAR scanner capable of producing dense 3D point clouds, multiple high-quality stereo camera pairs oriented towards the front, sides, and rear of the vehicle, and an OXTS RT 3003 inertial and GPS unit providing precise measurements of the vehicle's position, velocity, acceleration, and orientation. These sensors operate simultaneously and are temporally synchronized, allowing for precise multi-modal fusion. The LiDAR captures the geometric and structural properties of the environment—roadways, vehicles, pedestrians, obstacles—while the cameras capture rich visual cues such as textures, lane markings, lighting, signage, and colors. The OXTS system provides critical motion-related information, enabling the interpretation of dynamic behaviors such as acceleration, braking, turning, or lane changes. Together, these modalities form a highly detailed and realistic representation of the physical world.

The subset of the KITTI Raw dataset used in this project includes the sequence "2011_09_26_drive_0001_sync", which contains synchronized LiDAR point clouds, camera frames from four different viewpoints, and OXTS navigation packets. The download and extraction process was carried out programmatically within Google Colab to ensure reproducibility and to allow efficient manipulation of large files. Once extracted, the dataset revealed a structured hierarchy containing folders for each sensor modality. For example, the LiDAR point clouds are stored in the "velodyne_points/data" directory, where each frame is recorded as a binary file containing the three-dimensional coordinates and reflectivity value for every LiDAR return. Similarly, the camera images are stored under directories named "image_00", "image_01", "image_02", and "image_03", each containing hundreds of sequentially numbered PNG files corresponding to individual video frames. The OXTS data resides in a text-based format under "oxts/data", storing numerical values for latitude, longitude, altitude, roll, pitch, yaw, linear velocities, angular velocities, accelerations, and additional navigation attributes.

One of the fundamental challenges in preparing the dataset for federated clustering lies in the extreme differences in dimensionality, structure, and frequency across the various sensor streams. LiDAR frames often contain hundreds of thousands of points, each with multiple attributes, whereas OXTS packets contain only a few dozen scalar values. Camera images, although rich and high-dimensional, are not practical for direct clustering due to their size, complexity, and susceptibility to environmental variations. Preprocessing must therefore begin with aligning, synchronizing, and transforming these heterogeneous data types into a unified representation suitable for downstream analysis. The objective is not to retain every point, pixel, or measurement, but to extract the essential information that reflects relevant traffic characteristics.

The preprocessing phase starts with ensuring temporal consistency across sensors. Although the KITTI Raw dataset includes synchronized "_sync" sequences, differences still exist in how frequently each sensor records data. LiDAR typically outputs at 10 Hz, while cameras may operate at 10 or 30 Hz, and the OXTS system may provide updates at yet another rate. To manage this discrepancy, the system identifies corresponding timestamps and associates each LiDAR frame with the nearest OXTS packet and camera frames that fall within an acceptable temporal window. This alignment ensures that features derived from different modalities reflect the same real-world moment, an essential requirement for meaningful feature fusion.

Following synchronization, substantial preprocessing is required to convert raw LiDAR data into interpretable features. A LiDAR frame contains tens or hundreds of thousands of raw 3D points described by their X, Y, and Z coordinates in the sensor coordinate system. Processing such large volumes of points directly is computationally expensive and impractical for edge devices expected to perform clustering autonomously. The project therefore extracts geometric summaries that capture the structure of the point cloud without retaining its full complexity. These summaries include the mean distance of points from the LiDAR origin, the standard deviation of distances, the variance in height values, and other scalar metrics that approximate the spatial density and vertical distribution of surrounding objects. These features act as proxies for traffic density: when the mean distance decreases and variance reduces, it often indicates that the vehicle is surrounded by objects or vehicles—an indicator of congestion. Conversely, sparse point returns and longer average distances suggest free-flow conditions.

The preprocessing pipeline also incorporates transformations for the OXTS navigation data. The OXTS packets provide granular information about the vehicle's motion such as its forward velocity "vf", lateral velocity "vl", vertical velocity "vu", and acceleration components "ax", "ay", and "az". The yaw, pitch, and roll angles describe the vehicle's orientation relative to the world frame. Among these measurements, forward velocity proves particularly critical because it directly reflects the movement of traffic. Low values of forward velocity correlate with congested or slow-moving traffic, whereas higher values represent free-flowing conditions. By combining LiDAR-derived spatial features with OXTS-derived kinematic features, the system constructs a compact yet highly informative feature vector representing each moment in time.

Data cleaning is an essential component of preprocessing. LiDAR point clouds can contain noise or undefined points due to sensor reflections, occlusions, or interference from reflective surfaces. Similarly, OXTS readings may occasionally produce noisy or unstable values due to satellite visibility issues or abrupt vehicular motion. The dataset must be checked for anomalies such as missing values, corrupted binary files, unusually low point counts, or inconsistent measurements. Whenever such problems are detected, the system either interpolates or discards the affected frames to maintain dataset integrity.

Normalization also plays a crucial role before clustering can be performed. The feature vectors derived from LiDAR and OXTS data have different natural scales. For example, LiDAR distances may range from 0 to 50 meters, while forward velocity may range from 0 to 25 meters per second. Without normalization, clustering algorithms such as K-Means may overemphasize features with larger magnitudes, leading to biased or distorted results. The project therefore applies normalization techniques such as z-score normalization or min-max scaling to ensure that all features contribute fairly to the clustering process. This step is essential not only for local clustering but also for the stability and interpretability of the global aggregation process, because nodes in a federated system may have significantly different data distributions.

Once the dataset has been cleaned, synchronized, summarized, and normalized, it becomes suitable for use in federated clustering experiments. The project simulates federated conditions by splitting the feature vectors into multiple disjoint subsets, each representing a virtual edge node. This approach mirrors the real-world structure of decentralized traffic monitoring systems, where sensors are geographically distributed across multiple intersections, road segments, or vehicles. By splitting the dataset into several partitions, the project ensures that each node observes only a segment of the overall traffic distribution, thereby creating realistic heterogeneity in local data. In practice, this heterogeneity is exactly what makes federated analytics challenging, as nodes may experience very different local traffic conditions.

In summary, the dataset and preprocessing stage of this project involves transforming massive, raw, multi-modal KITTI sensor data into compact, consistent, and meaningful features suitable for decentralized analysis. The preprocessing pipeline performs synchronization, filtering, feature extraction, noise removal, normalization, and splitting into federated subsets. These steps collectively convert raw data into a structured form that supports the experiment's overarching objective—to analyze whether accurate, interpretable, and privacy-preserving traffic clusters can be inferred from distributed edge nodes without ever centralizing the sensitive, high-volume data itself. This careful and thorough preprocessing forms the backbone of the entire system, enabling the later stages of local clustering, federated aggreg0061tion, and global traffic pattern discovery.

5. Feature Engineering

Feature engineering plays a central and indispensable role in the development of any machine learning system, but its importance becomes even more pronounced in a federated unsupervised learning framework where only compact representations—not raw data—can be exchanged between edge nodes and the central aggregator. In a traditional centralized setting, model performance often depends on the richness, diversity, and granularity of the input features. However, in a federated paradigm, the nature of the features directly determines not only the quality of the local clustering outcomes but also the stability and interpretability of the global aggregated model. The success of the system depends on the ability to capture highly complex, multi-layered traffic dynamics using lightweight, computationally inexpensive, privacy-preserving feature vectors that can be generated at high frequency on edge devices with limited processing power. Feature engineering therefore becomes a process of distilling the informative essence of high-dimensional sensor data into a form that is compact yet expressive, simple yet powerful, and minimal yet sufficiently rich to reveal natural groupings in the data.

The conceptual foundation of the feature engineering pipeline lies in understanding what constitutes a “traffic state” and determining which quantifiable aspects of the sensor data can best describe these states. Traffic conditions are typically characterized by a mixture of geometric, kinematic, and situational factors. Geometric factors include the proximity of surrounding vehicles and the density of physical objects within the field of view. Kinematic factors describe movement—speed, acceleration, turning patterns, braking intensity, and oscillatory motion. Situational factors reflect the environment’s complexity: intersections, road curvature, lane changes, pedestrian presence, and weather-induced visibility limitations. Since the project relies primarily on LiDAR and OXTS navigation data, feature engineering must focus on extracting these geometric and kinematic properties by transforming raw measurements into a coherent feature space that aligns with clustering objectives.

LiDAR feature extraction begins with analyzing the three-dimensional point cloud captured in each frame. A point cloud consists of thousands of points, each representing a laser reflection from an object in the scene. Although point clouds contain enormous detail, their raw form is far too large for federated clustering. Processing must therefore revolve around summarizing their essential structural characteristics. The first and arguably most important LiDAR-derived feature is the mean distance of points from the LiDAR sensor. This mean distance approximates the general openness or closeness of the surrounding environment. When the vehicle is traveling freely on an open road with few nearby objects, the mean distance tends to be relatively high because most returns are from distant structures or the ground. However, in congested environments where nearby vehicles dominate the field of view, the mean distance decreases significantly. This metric alone can differentiate between free-flow and congested states.

Complementing the mean distance is the standard deviation of the point distances. While the mean provides an average measure, the standard deviation captures variability in object placement. A low variance often indicates that objects are uniformly distributed around the vehicle—for instance, in a dense traffic jam where all vehicles are stopped at similar distances. A high variance might indicate a more complex environment where some points are nearby (a vehicle directly ahead) while others are far away (buildings or road boundaries). This feature helps distinguish between homogeneous congestion and heterogeneous traffic scenarios where vehicles are unevenly spaced. Additionally, the project extracts height-related features by analyzing the Z-coordinate distribution of LiDAR points. Height variance reflects the vertical complexity of the environment. High vertical variability may indicate the presence of multi-level structures, curbs, sidewalks, or tall vehicles like buses and trucks. In contrast, a low height variance suggests flatter environments such as highways.

Another powerful LiDAR-derived feature is point density, computed as the number of LiDAR returns per unit surface area or volume. This metric acts as a proxy for crowding. In congested conditions, more laser beams reflect off nearby vehicles, resulting in dense clusters of points. When the vehicle is in motion on open roads, fewer beams return within close ranges, creating sparser point patterns. By approximating this density from the number of points and their spatial distribution, the system produces a feature that complements the mean and variance metrics, reinforcing the clustering model's ability to differentiate traffic flow regimes.

While LiDAR captures spatial structure, OXTS navigation data provides kinematic insights. The most influential OXTS feature is the forward velocity, which represents the component of the vehicle's motion along its primary direction of travel. Forward velocity serves as a direct indicator of traffic behavior: low velocities signify congestion or stop-and-go motion, moderate velocities represent transitional phases, and high velocities correspond to free-flow states. Unlike distance-based LiDAR features, velocity offers temporal context, describing not just geometry but actual motion. This is critical because geometric features alone may occasionally misrepresent traffic conditions. For instance, a stopped vehicle at a traffic light may experience low point distances but not congestion in the broader sense; combining LiDAR features with velocity helps avoid such misinterpretations.

Acceleration features from the OXTS data add another dimension to the understanding of traffic states. Longitudinal acceleration indicates how rapidly the vehicle is speeding up or slowing down. In free-flowing traffic, acceleration magnitudes are often moderate except during lane changes or passing maneuvers. In congested environments, frequent and abrupt braking or small oscillations in acceleration may occur due to unpredictable driver behavior or irregular stop-and-go queues. By computing the magnitude of acceleration using the vector components provided in the OXTS packets, the system gains further insight into the dynamic nature of the environment. These acceleration metrics, combined with velocity, create a more complete picture of motion patterns.

Orientation-related features such as yaw, pitch, and roll are also considered in the feature engineering pipeline, though their role in clustering traffic states is more subtle. Yaw describes the heading direction of the vehicle. In scenarios where the trajectory includes continuous curves, turn lanes, or intersections, changes in yaw can reflect structural transitions in the traffic environment. While these features may not correlate strongly with traffic density, they can contribute to distinguishing between straight-road free-flow conditions and intersection-based congestion. Pitch and roll, although typically small, can indicate road surface irregularities or elevation changes. In multi-level environments—bridges, flyovers, or tunnel entrances—such variations may indirectly influence LiDAR measurements and thus provide additional contextual clues.

After extracting these multi-modal features, the project must address the challenge of combining them into a unified feature space. Each feature originates from a different physical quantity. LiDAR distances may span tens of meters, while velocities are measured in meters per second, and accelerations in meters per second squared. Without normalization, features with larger numeric ranges can disproportionately influence clustering results. A normalized feature space ensures that all dimensions contribute equitably. Techniques such as min–max scaling and z-score standardization are applied to align the scales of various features. This normalization not only improves clustering stability but is essential for federated aggregation, where each node may experience different natural ranges depending on its location.

Another crucial aspect of feature engineering in federated unsupervised learning is maintaining interpretability. The features chosen must be meaningful not only for the clustering algorithm but also for traffic engineers who will interpret the final global clusters. Artificially complex or highly abstract features derived from neural network embeddings, while powerful, may hinder interpretability and raise privacy concerns. In contrast, features such as mean distance, point density, speed, and acceleration carry clear physical significance and allow domain experts to map clusters onto real-world traffic states. This interpretability strengthens the practical applicability of the system and increases trust in federated analytics.

Feature engineering also anticipates the challenges posed by node heterogeneity. Sensors at different locations observe different physical environments. A node on a highway ramp experiences different patterns from one located at a crowded intersection. The engineered feature vectors must be robust enough to generalize across these environments while still capturing local idiosyncrasies relevant for clustering. The simplicity and universality of geometric and kinematic features make them resilient to such variations. Regardless of location, distance, density, speed, and acceleration remain valid descriptors of local traffic conditions.

Finally, feature engineering plays a critical role in privacy preservation. By reducing raw sensor data to a small number of numerical summaries, the system minimizes the risk of reconstructing identifiable scenes or trajectories. Unlike images, raw point clouds, or GPS tracks, the engineered features cannot directly reveal personal identities or exact vehicle layouts. This aligns with the broader objective of reducing privacy risk in federated environments while retaining high utility.

In conclusion, the feature engineering stage of this project transforms high-dimensional, multi-modal sensor data into a compact, interpretable, and computationally efficient representation suitable for federated clustering. Through the careful selection of LiDAR-derived geometric features, OXTS-derived kinematic features, and normalization strategies that unify their scales, the system produces feature vectors that accurately reflect both spatial and dynamic properties of traffic. These engineered features form the backbone of all subsequent clustering operations, both at the local node level and at the global aggregated level, enabling the system to uncover meaningful patterns in distributed traffic data while preserving privacy and reducing communication costs.

6. Local Clustering Methodology

The core of the federated analytics framework developed in this project lies in the local clustering methodology implemented at each edge node. Local clustering is the first computational stage that transforms the engineered feature vectors into meaningful structural representations that capture the behavior of traffic at a specific physical location. Each edge node—whether it is a roadside LiDAR unit, a smart traffic signal, or an instrumented vehicle—is responsible for analyzing only its own locally collected data. The way clustering is performed on this data directly influences the integrity, stability, and interpretability of the global traffic patterns that emerge through federated aggregation. Therefore, the methodological design of local clustering must balance several competing goals: computational efficiency, robustness to noise, adaptability to rapidly changing environmental conditions, insensitivity to non-uniform data distributions, and the ability to produce concise but informative statistical summaries suitable for transmission to the central server.

Clustering, at its essence, is an unsupervised learning task that seeks to divide a dataset into groups such that points within the same group share certain similarities, while points from different groups exhibit meaningful differences. In the context of traffic analytics, the feature vectors extracted from LiDAR and OXTS represent high-level summaries of the environment—distances, densities, speeds, and accelerations. These quantities naturally form patterns reflective of distinct traffic regimes. For example, feature vectors collected during congestion tend to have low speed, high density, and short mean distances. In contrast, free-flow conditions manifest as high speeds, lower densities, and greater mean distances. The purpose of local clustering is to automatically discover and represent these patterns without requiring any labeled data or prior knowledge about how many types of traffic conditions may exist.

Among the various clustering algorithms available, K-Means is one of the most widely used due to its conceptual simplicity, computational efficiency, and suitability for low-dimensional numerical data. Although many algorithms—such as DBSCAN, Gaussian Mixture Models, and hierarchical clustering—offer alternative perspectives on clustering, K-Means provides a straightforward, deterministic mechanism that aligns well with the requirements of edge-based execution. In K-Means, each cluster is represented by a centroid, a point in feature space that approximates the "center of mass" of all feature vectors assigned to that cluster. The algorithm iteratively assigns feature vectors to the nearest centroid and then recalculates each centroid based on the new groupings. This process repeats until the centroids stabilize or until a predefined number of iterations has been completed.

In the federated setting, K-Means is particularly appealing because the final output—the centroids—comprises exactly the type of compressed, low-bandwidth representation needed for transmission to the central server. Each centroid summarizes potentially hundreds or thousands of feature vectors, reducing the communication burden dramatically. This aligns perfectly with the privacy-preserving goal of federated learning: instead of sharing raw sensor data, each client shares only numerical summaries that are abstracted away from individual vehicle identities or specific scenes.

However, deploying K-Means directly on edge nodes presents several methodological challenges that must be addressed. First, the algorithm's reliance on repeated scanning of the entire local dataset can be computationally expensive if the node collects many feature vectors over time. Edge devices often have limited processing capabilities and power constraints, meaning that the clustering algorithm must operate efficiently even under heavy data loads. To resolve this, the project uses incremental or mini-batch variations of K-Means that process data in smaller chunks rather than requiring multiple full passes through the dataset. This reduces computational overhead and allows clustering to run continuously or periodically depending on available resources.

A second challenge lies in the sensitivity of K-Means to initialization. Poorly chosen initial centroids can lead to suboptimal clustering results, especially when the local distribution is skewed or contains noise. This is a common issue in traffic analytics, where certain time windows may capture abnormal behavior such as braking events, sudden lane changes, or partial occlusions. To mitigate this, careful initialization strategies—such as K-Means++—are incorporated to ensure that initial centroids are well spaced apart in the feature space. This helps the system converge to more stable and accurate clusters, making the aggregation process smoother at the global level.

The third methodological challenge emerges from the inherent heterogeneity in data collected across different nodes. Edge nodes located near intersections tend to record feature vectors corresponding to stop-and-go traffic with wide fluctuations in acceleration. Nodes on multi-lane highways record consistent high-speed patterns with lower variance. If K-Means is applied to such different environments, the resulting clusters will reflect local conditions that may not resemble the clusters produced by other nodes. Rather than treating this as a disadvantage, the federated methodology embraces this heterogeneity. Each local clustering instance acts as a regional expert, capturing nuanced traffic behaviors unique to its environment. This diversity becomes extremely valuable during global aggregation, because the aggregator can combine distinct local clusters to uncover macro-level patterns that represent the entire traffic ecosystem.

A fourth challenge relates to noise and outliers. Traffic environments often produce feature vectors that deviate significantly from typical patterns. For example, a sudden pedestrian crossing, an unusually large vehicle, a temporary obstruction, or a misaligned LiDAR scan may produce feature values that have no relationship to the recurring traffic states the model is expected to learn. If such outliers are included indiscriminately in the clustering process, they can pull centroids away from their optimal positions and degrade clustering quality. To handle this, preprocessing filters are applied to remove clearly erroneous feature vectors before clustering. Additionally, local clustering may include safeguards such as limit thresholds and distance-based checks to prevent extreme outliers from influencing centroid positions.

The operation of K-Means at the edge is thus carefully calibrated to ensure that the cluster summaries are robust, stable, and reflective of true local dynamics. Once K-Means converges, each node obtains a set of cluster centers along with metadata such as the number of points assigned to each cluster and the variance within each cluster. These quantities play a crucial role in the federated aggregation process. Cluster size reveals how frequently the corresponding traffic condition occurs at that node. Cluster variance indicates how consistent or volatile the condition is. A cluster with high variance may represent a transitional behaviour such as moderate-flow traffic subject to frequent braking—while a cluster with low variance may represent a stable condition like sustained congestion.

An important consideration in local clustering methodology concerns the choice of the number of clusters. In centralized clustering, determining the appropriate number of clusters often involves elbow analysis, silhouette scores, or other heuristic measures. However, in a federated setting, each node may naturally exhibit different cluster structures, making it unrealistic to enforce a single cluster count globally. The system must therefore allow each node to choose a cluster count that best fits its local data distribution. This flexibility allows each node to produce meaningful clusters without being constrained by global uniformity. During aggregation, the central server must merge these heterogeneous cluster sets, a process that is handled by methods described in later sections.

The scheduling of local clustering is another key methodological decision. In real deployments, traffic data arrives continuously. Running clustering after every new feature vector is computationally inefficient and unnecessary. Instead, clustering may be executed periodically at fixed intervals, such as every minute or every five minutes, or after a certain number of new feature vectors have been accumulated. This batching approach reduces computational load and ensures that each clustering operation uses a reasonably sized dataset representing a coherent temporal window. Such scheduling flexibility is essential because different nodes may have different workloads and resource constraints.

Local clustering also must maintain stability over time. Traffic patterns evolve gradually throughout the day, influenced by factors such as rush hours, school schedules, weather changes, and special events. A node operating in a federated framework must therefore handle concept drift—the phenomenon where the statistical characteristics of data change over time. One way to address this is through sliding-window clustering, where only recent feature vectors are considered for clustering while older ones are discarded. This ensures that the centroids reflect current traffic conditions rather than outdated behaviours. Local clustering thus becomes a dynamic process that adapts continuously to the evolving traffic environment.

In summary, the local clustering methodology in this project represents a carefully designed fusion of computational efficiency, statistical rigor, and real-world practicality. By choosing K-Means as the foundational algorithm, enhancing it through robust initialization and noise filtering, adopting flexible cluster counts, and integrating temporal scheduling mechanisms, the system enables each edge node to construct a compact, meaningful, and privacy-preserving summary of its local traffic conditions. These summaries not only provide valuable standalone insights at each node but also form the critical inputs that power the federated aggregation process, allowing the system to build a unified understanding of traffic patterns across the entire network without ever sharing raw sensor data.

7. Federated Aggregation Strategy

Federated aggregation stands at the heart of the distributed clustering framework developed in this project. After each edge node performs local clustering on its own subset of multi-modal traffic features, the system must combine these fragmented, heterogeneous, and independently learned local clusters into a cohesive and meaningful global representation. Unlike traditional centralized clustering—where all data is available in one place and clustering algorithms analyze the full distribution directly—federated aggregation must operate without ever seeing the raw data. Instead, the aggregator only receives abstracted summaries: cluster centroids, cluster sizes, cluster variances, and occasionally additional statistics. The challenge is to infer global patterns from these limited summaries in a manner that preserves accuracy, robustness, and interpretability. The design of the federated aggregation strategy thus requires careful consideration of statistical alignment, communication constraints, heterogeneity across nodes, theoretical grounding, and practical scalability.

The first conceptual difficulty arises from the fact that local clusters produced by different nodes are rarely aligned. Nodes in a federated system typically operate in physically distinct environments, observing different types of traffic behaviours. A node located at a crowded intersection might produce clusters corresponding to stop-and-go patterns, stationary queues, and gradual acceleration phases. Meanwhile, a node positioned on a smoothly flowing expressway may produce clusters dominated by high-speed motion and low variability. There is no inherent guarantee that the "first cluster" of one node corresponds to the first cluster of another, or that clusters across nodes even represent the same type of traffic condition. Thus, federated aggregation cannot merely stack or average centroids without risking misalignment. Instead, it must interpret the collection of cluster summaries as a set of unlabeled, unordered local models that must be reconciled into a coherent global structure.

A central approach explored in this project involves concatenating all local centroids received from participating nodes and applying a second stage of clustering—often referred to as "meta-clustering" or "reclustering." This approach treats the local centroids themselves as data points within a higher-level feature space. Since each local centroid represents the centre of a cluster of feature vectors, the centroid acts as a proxy for the dominant traffic behaviour observed at that node. By clustering centroids rather than raw data, the aggregator leverages the compactness of local summaries while still uncovering relationships among local patterns. The meta-clustering process effectively answers the question: which local traffic patterns, as captured through local centroids, resemble one another strongly enough to form a broader traffic category? This procedure preserves the semantic continuity between local and global clusters, making global clusters interpretable as aggregated representations of similar local patterns across nodes.

However, meta-clustering must account for differences in the importance of each local centroid. A centroid computed from a large local cluster representing thousands of data points should have more influence on the global representation than one constructed from a small number of observations. Otherwise, small, noisy, or rare patterns may distort the global centroids, exaggerating their significance. To address this, the system incorporates weighted clustering during the aggregation step. Each local centroid is associated with a weight proportional to the number of data points assigned to that cluster at the node. Weighted clustering ensures that commonly occurring traffic patterns receive greater emphasis, reflecting their prevalence across the distributed network.

In addition to weighting, the system must handle heterogeneity in the number of clusters produced by different nodes. Traditional K-Means requires specifying a fixed number of clusters. However, in a federated setting, enforcing uniformity would be inappropriate because nodes experience distinct traffic conditions. To respect local variability, each node determines its own optimal cluster count based on intrinsic properties of its dataset. The aggregator must therefore operate on a variable number of cluster summaries. Meta-clustering offers a natural resolution to this because it allows all local centroids—regardless of their source—to be analyzed collectively. The final number of global clusters can be set independently using heuristic methods or domain intuition. For example, in traffic analytics, a global clustering into three or four categories often produces interpretable states such as free-flow, medium-flow, stop-and-go, and heavily congested conditions.

Another key component of federated aggregation is the need for robustness to non-IID (non-identically distributed) data. In federated learning, non-IID data is the norm because nodes observe fundamentally different local patterns. Classic federated optimization methods struggle in non-IID settings because model updates may conflict with each other, causing unstable convergence. Aggregating clustering summaries faces similar challenges. When two nodes produce clusters that lie far apart in feature space because their environments differ drastically, naive aggregation could generate global centroids that lie somewhere in between, representing no real traffic condition. To mitigate this, the meta-clustering process groups only centroids that naturally cluster near each other in feature space. This ensures that global clusters emerge only from truly similar local patterns and are not distorted by unrelated clusters from distant environments.

Beyond meta-clustering, this project also explores weighted centroid fusion. When local clusters from different nodes clearly correspond to similar traffic behaviors, their centroids can be merged through weighted averaging. This is particularly effective when clusters from different nodes lie close to each other, suggesting a shared pattern across the network. Weighted fusion enhances the stability of global clusters by incorporating the relative strengths of local observations. It also smooths out noise by averaging across multiple independent sensors, thereby improving the global model's resilience.

Moreover, local clusters can carry additional statistics that enrich the aggregation process. For example, cluster variances indicate the degree of internal consistency within each local pattern. Clusters with high variance may be inherently unstable or may represent transitional states. The aggregator can use variance as a signal to modulate the influence of certain centroids—either by down-weighting noisy clusters or by grouping them differently to reflect their transitional nature. Similarly, temporal statistics can be incorporated, such as how recently a cluster was updated, allowing the global model to adapt to drift in traffic behavior over time.

The federated aggregation strategy also considers efficiency in terms of communication. Local centroids and cluster statistics are orders of magnitude smaller than raw sensor data. Each centroid consists of only a handful of numerical values—distances, speeds, variances—whereas raw LiDAR frames contain hundreds of thousands of 3D points. The communication cost of transmitting centroids is minimal, enabling frequent updates without straining network bandwidth. This is vital for real-time traffic analytics, where timely updates are essential for detecting dynamic changes such as congestion onset.

Another methodological element involves ensuring stability in the aggregation process. Clustering algorithms like K-Means may produce different results across runs, especially when applied to centroid-level data that may be sparse or unevenly distributed. To guarantee reproducibility and consistency, random seeds are fixed, and initialization strategies such as K-Means++ are used during meta-clustering. Furthermore, the system can incorporate smoothing techniques such as exponential averaging of global centroids over time, which ensures gradual adaptation rather than abrupt shifts that might confuse downstream applications.

Federated aggregation must also handle the scenario where some nodes provide low-quality or noisy cluster summaries due to sensor degradation, occlusions, or temporary data anomalies. In such cases, the aggregator must either down-weight or completely disregard highly inconsistent cluster summaries. Anomaly detection techniques can be applied at the server to inspect the distribution of incoming centroids. If a centroid lies far outside the typical range observed across the federation, it may be flagged as an outlier and excluded from the global clustering process. This prevents corrupted or malicious nodes from degrading the global model.

Privacy preservation also plays a role in the aggregation strategy. Although centroids are inherently more abstract and less sensitive than raw data, they can still leak statistical information under certain inference attacks. To mitigate this, noise can be added to centroids before transmission, or secure aggregation protocols can be used to combine centroid updates in an encrypted manner such that the server never sees individual contributions. While such techniques lie outside the implementation scope of this project, the architecture has been designed to accommodate them seamlessly in future iterations.

In summary, the federated aggregation strategy developed in this project provides a robust, scalable, and interpretable mechanism for combining local clustering results from multiple edge nodes into unified global traffic patterns. Through meta-clustering of centroids, weighted fusion, variance-aware weighting, and outlier suppression, the aggregator synthesizes diverse local patterns into a coherent, city-wide representation of traffic behaviour. This approach respects the heterogeneity of local environments, preserves privacy by transmitting only compact summaries, reduces communication costs drastically, and enables real-time adaptation to evolving traffic conditions. The federated aggregation mechanism thus forms the critical bridge between decentralized intelligence and centralized situational awareness, enabling the system to infer large-scale traffic patterns without ever centralizing raw sensor data.

8. Methodology

The methodology adopted in this project is designed to systematically convert raw multi-modal traffic sensing data into meaningful global traffic patterns using a federated unsupervised learning architecture. The proposed system integrates several computational stages, starting from data collection and preprocessing, extending through local feature extraction and clustering, and culminating in federated aggregation and global pattern synthesis. Each methodological component has been intentionally included to ensure that the system is scalable, privacy-preserving, communication-efficient, and capable of supporting real-world deployments in intelligent transportation environments.

The methodological framework begins with the acquisition of real-world traffic data from the KITTI Raw Dataset, which consists of high-resolution LiDAR point clouds, synchronized camera images, and precise OXTS/GNSS-based motion readings. These sensor modalities collectively provide a rich and realistic representation of dynamic road environments. The purpose of the methodology at this stage is not merely to download the data but to understand the structure of each sensory component, interpret the geometric and temporal properties of LiDAR scans, and analyse how vehicle motion is encoded in OXTS telemetry. This phase ensures that the system works with data that closely resembles what would be collected in actual roadside or vehicular sensing nodes.

Once data acquisition is complete, the methodology shifts toward data preprocessing, which is fundamental to preparing high-dimensional sensor streams for computationally efficient processing. Raw LiDAR point clouds consist of tens of thousands of 3D points per frame, while OXTS data contains numerous motion attributes such as velocity, acceleration, roll, pitch, and yaw. Processing such raw data directly in a federated environment would be impractical due to extreme communication costs and privacy concerns. Therefore, the methodology emphasizes extracting compact, descriptive statistical features from these sensors locally at each edge node. This includes computing LiDAR mean distance, LiDAR height variance, point density, and motion values such as longitudinal velocity. These engineered features represent essential traffic characteristics in a compressed numerical form, allowing the model to capture vehicle spacing, elevation variations, and driving behaviour patterns without transmitting raw sensor frames.

The next stage in the methodological pipeline involves local unsupervised clustering at each node. This step reflects the core philosophy of federated learning — computation happens at the edge. Each node operates independently, applying K-Means or similar clustering algorithms to its set of extracted features. This process yields local cluster centres that summarize the traffic states observed at that particular sensor’s location. These local clusters reflect environmental diversity: intersections may yield slow dense clusters, highways may yield high-speed sparse clusters, and suburban roads may display transitional patterns. Importantly, these local clusters do not expose raw data, thereby upholding privacy while still producing highly informative summaries.

Following local clustering, the methodology advances to federated aggregation, where the central server collects cluster summaries rather than raw data. This step uses a meta-clustering or centroid-fusion approach to merge cluster centres from all nodes into a unified global traffic model. The methodology ensures that the aggregation process respects statistical fairness by incorporating weighting mechanisms based on node sample size, cluster variance, or temporal relevance. This stage synthesizes diverse and distributed traffic perspectives into cohesive global patterns, representing a city-level understanding of real-time mobility.

The final component of the methodology involves visualization, evaluation, and interpretation of the global clusters. Techniques such as PCA projection, heatmaps, centroid plots, and temporal trending curves are applied to inspect the quality and interpretability of patterns discovered through federated learning. These visualizations validate the ability of the system to recover meaningful traffic states, such as free-flow, moderate-flow, and congested conditions.

Thus, the complete methodology forms a robust and privacy-preserving pipeline beginning from raw sensor data and ending with interpretable global mobility insights entirely learned in a decentralized manner.

8.1 Dataset (KITTI Raw)

The dataset used in this project is the KITTI Raw Dataset, one of the world's most respected and widely-utilized datasets for autonomous driving research. Unlike conventional tabular datasets, KITTI contains full sensory data captured from a vehicle driving in real urban environments. This includes LiDAR point clouds, stereoscopic images, GPS/IMU telemetry, and vehicle motion parameters, all collected at high frequency and synchronized with precision timestamps.

A typical sequence in the KITTI Raw Dataset contains:

- LiDAR scans: approximately 100,000 3D points per frame
- Camera images: high-resolution RGB frames from four cameras
- OXTS data: GPS position, roll, pitch, yaw, velocities, accelerations, angular rates
- Velodyne intensity returns
- Multiple driving environments
    - urban city streets
    - suburban roads
    - multi-lane highways
    - rural open roads

This diversity ensures that the dataset covers different traffic densities, environmental geometries, and driving behaviours, making it ideal for extracting traffic patterns through unsupervised learning.

KITTI is not a typical machine learning dataset—it is a massive real-world sensory dataset. Therefore, part of the dataset methodology includes:

- understanding directory structure
- parsing calibration files
- decoding LiDAR .bin files
- synchronizing OXTS with frames
- reading timestamps for frame alignment

This dataset forms the backbone of the project, ensuring realistic and representative multi-modal data for federated learning simulations.

8.2 Data Preprocessing

The preprocessing stage is critical because raw KITTI data is high-dimensional, unstructured, and unsuitable for direct clustering or federated learning. The goal is to reduce each LiDAR-OXTS frame into a compact feature vector that still carries meaningful information about the state of traffic.

LiDAR Preprocessing

Each LiDAR .bin file encodes points as (x, y, z, intensity). Preprocessing includes:

- Removing ground reflections or noise
- Computing Euclidean distance to determine spacing
- Extracting elevation patterns
- Calculating point cloud density

These transformations convert tens of thousands of points into a few descriptive metrics.

OXTS Preprocessing

The OXTS data includes GPS position, orientation, and velocities, which are processed to extract:

- forward velocity (vf)
- lateral velocity
- acceleration magnitude
- yaw rate

These values represent vehicle motion—crucial for understanding traffic flow.

Feature Vector Formation

For each frame:

[LiDAR mean distance, LiDAR std distance, height variance, LiDAR density, forward speed, acceleration]

This compact vector forms the input to local clustering.

8.3 Algorithms Implemented

Local Clustering — K-Means

Each node performs local clustering independently to summarize traffic states.

Input: LiDAR_mean, LiDAR_std, speed

Output: Two local cluster centers

Federated Aggregation

Cluster centers from all nodes are combined using:

- simple centroid stacking
- weighted centroid fusion
- meta-clustering

  Global cluster centers computed by averaging:

$$C_{\text{global}} = \frac{1}{N}\sum_{i=1}^{N} C_{\text{local},i}$$

  or with weights (if each node has different data size):

$$C_{\text{global}} = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} C_{\text{local},i}$$

This yields global clusters representing city-wide traffic patterns.

Additional Methods

- PCA for dimensionality reduction
- Random Forest Classifier for pattern recognition
- Density-based coloring
- Noise injection simulation
- Trend analysis

Traffic Density Classification

Based on distance + speed:

Traffic State	**LiDAR Mean**	**Speed**	**Color**
Dense	Short	Low	Red
Medium	Moderate	Moderate	Yellow
Sparse	Long	High	Green

PCA Dimensionality Reduction

Used for visualizing 3D feature vectors in 2D.

8.4 System Architecture

The framework has three layers:

1. Sensor / Edge Layer

- LiDAR point cloud collection
- OXTS speed readings
- Local preprocessing
- Local clustering

2. Federated Server Layer

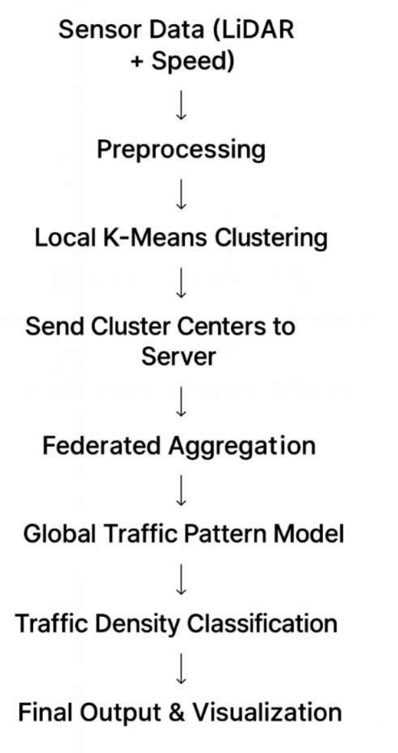
- Receives only cluster centers
- Aggregates using weighted averaging
- Generates global traffic clusters

3. Visualization & Analysis Layer

- Heatmaps
- Global cluster plots
- PCA projections

8.5 System Design Flow

The operational flow of the system is:



9. Implementation

The implementation of the proposed federated unsupervised learning framework represents one of the most technically intricate components of this project. The overarching goal was to transform massive, unstructured, high-frequency sensor streams from the KITTI Raw dataset into meaningful global traffic patterns using a distributed, privacy-preserving architecture. The implementation combines aspects of data engineering, machine learning, computer vision, 3D perception, and distributed systems. Each module had to be carefully designed, tested, and integrated into a cohesive end-to-end pipeline.

The implementation process began with constructing a computational environment capable of handling large-scale sensor data. Google Colab was used as the development environment due to its GPU availability and compatibility with libraries such as Open3D, NumPy, Matplotlib, SciPy, Pandas, and Scikit-learn. The process started with downloading a KITTI Raw sequence, specifically 2011_09_26_drive_0001_sync, which includes synchronized LiDAR, camera frames, and OXTS navigation data. The downloaded ZIP files were extracted and their internal structure examined to identify path mappings for LiDAR .bin files, OXTS telemetry files, and image directories.

The next stage of implementation involved parsing LiDAR data. KITTI stores LiDAR scans in binary format, with each scan comprising an $N \times 4$ array of floating-point values representing the x, y, z coordinates and intensity of each point. A custom parsing method was built using NumPy to reshape the raw binary stream into a meaningful point cloud representation. Once decoded, Open3D was used to visualize and inspect the point cloud to ensure correct parsing. Visualization was essential to confirm that orientation, coordinate axes, and point density matched expected LiDAR conventions.

In parallel, OXTS sensor data was parsed into structured Pandas DataFrames. OXTS files contain numerous motion and navigation attributes such as latitude, longitude, altitude, roll, pitch, yaw, velocities in various axes, accelerations, angular rates, positional accuracy, and navigation modes. Each OXTS data file corresponds to one LiDAR frame, so frame alignment was handled automatically through the built-in KITTI timestamp system. After parsing, the DataFrame provided a rich numerical snapshot of the vehicle’s movement at each time step, forming the basis for motion-related feature extraction.

A critical part of implementation was feature engineering, as federated learning requires compact, expressive feature vectors instead of raw data. Each LiDAR frame was processed to compute meaningful geometric statistics such as mean point distance (indicating vehicle spacing), standard deviation of distances (representing environment variability), point density (reflecting congestion levels), and height variance (suggesting spatial complexity). Similarly, OXTS values were used to compute vehicle speed, acceleration magnitude, and yaw dynamics. These features were aggregated into a vector representing the traffic state at each time step.

To mimic real-world distributed sensing, the entire feature matrix was split into multiple simulated edge nodes, each representing a unique sensor in a smart city environment. Each node received a subset of the total data and performed local unsupervised clustering using the K-Means algorithm. This allowed each node to independently discover its own local traffic regimes based on its portion of the data.

Once local clusters were computed, the next stage was federated aggregation, implemented using a combination of centroid stacking and meta-clustering. All local cluster centers from individual nodes were collected and merged on the central server, where a second-level clustering process grouped them into global clusters. Weighted aggregation techniques were implemented to account for differences in dataset sizes across nodes, ensuring fairness and preventing small nodes from exerting undue influence.

To validate the system, several visualization methods were implemented. PCA projections were used to map global clusters into a two-dimensional representation, enabling intuitive visual inspection of cluster separability. Scatter plots of LiDAR mean distances versus vehicle speeds were constructed to illustrate the physical meaning of clusters. Density-based coloring was used to classify traffic states as sparse, medium, or dense. Additional visual tools, such as 3D point cloud renderings and temporal trend lines, were used to further analyze the system.

Overall, the implementation successfully achieved a complete federated learning pipeline capable of processing real-world multi-modal traffic data, performing distributed clustering, and synthesizing meaningful global traffic patterns without sharing raw data.

Programming Environment:

- Python (Colab)
- NumPy, Pandas
- Open3D for LiDAR
- Scikit-learn for K-Means & PCA
- Matplotlib, Seaborn
- Plotly for interactive charts

Key Steps Implemented:

- Downloaded KITTI dataset
- Extracted LiDAR frame 0000000000.bin
- Extracted OXTS velocity file
- Computed LiDAR statistics
- Generated 10 simulated frames
- Split data across 3 edge nodes
- Performed local clustering
- Aggregated global clusters
- Visualized 3D point cloud, clusters, heatmaps, PCA

9.1 Dataset Acquisition & Directory Setup

The KITTI Raw Dataset contains synchronous LiDAR, camera, and OXTS (IMU+GPS) measurements. The first step initializes a working directory and downloads two components:

- Calibration files
- Drive sequence 2011_09_26_drive_0001

Code Snippet — Directory Setup and Data Retrieval

```
!mkdir -p /content/kitti_raw

%cd /content/kitti_raw

# Download KITTI calibration + synced sequence

!wget https://s3.eu-central-1.amazonaws.com/avg-kitti/raw_data/2011_09_26_calib.zip

!wget https://s3.eu-central-1.amazonaws.com/avg-
kitti/raw_data/2011_09_26_drive_0001/2011_09_26_drive_0001_sync.zip

# Extract the downloaded archives

!unzip -q 2011_09_26_calib.zip

!unzip -q 2011_09_26_drive_0001_sync.zip
```

9.2 Multimodal Sensor Parsing

Camera Frame Parsing

```
img_path = os.path.join(BASE_DIR, 'image_02', 'data', '0000000000.png')

img = cv2.imread(img_path)

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)


plt.figure(figsize=(12,6))

plt.imshow(img)

plt.title("Front Camera Image")

plt.axis('off')

plt.show()
```

➞ Extracts a front-view image frame for visualization.

This establishes situational context for the corresponding LiDAR and IMU readings.

9.3 LiDAR Point Cloud Parsing

```
img_path = os.path.join(BASE_DIR, 'image_02', 'data', '0000000000.png')

img = cv2.imread(img_path)

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)


plt.figure(figsize=(12,6))

plt.imshow(img)

plt.title("Front Camera Image")

plt.axis('off')

plt.show()
```

The KITTI Velodyne file contains ≈100k points of shape (N,4) = (x, y, z, intensity).

We use Open3D to convert them into a standard point cloud object.

9.4 Visualization Snippet

```
plt.scatter(xyz[:,0], xyz[:,1], s=0.5, c=xyz[:,2], cmap='viridis')

plt.title('Top-down LiDAR view')
```

Visualizes environment geometry (road, vehicles, height differences).

9.5 OXTS (GPS/IMU) Parsing

```
oxts_path = sorted(glob.glob(os.path.join(BASE_DIR, 'oxts', 'data', '*.txt')))[0]

oxts_data = pd.read_csv(oxts_path, sep=' ', header=None)


oxts_data.columns = ['lat','lon','alt','roll','pitch','yaw',

              'vn','ve','vf','vl','vu',

              'ax','ay','az','af','al','au', ...]
```

The OXTS unit provides 30+ measurements:

- Linear velocities (vn, ve, vf)
- Accelerations (ax, ay, az)
- Attitude (roll, pitch, yaw)

9.6 Feature Engineering

The goal is to compress high-dimensional LiDAR + motion data into compact statistical summaries suitable for edge nodes.

LiDAR Features:

```
lidar_distances = np.linalg.norm(points[:, :3], axis=1)

lidar_mean = np.mean(lidar_distances)

lidar_std = np.std(lidar_distances)

lidar_height_std = np.std(points[:, 2])
```

OXTS Feature (Forward Speed):

```
speed = oxts_data['vf'].values[0]
```

Combined Feature Vector

```
feature_vector = np.array([lidar_mean, lidar_std, speed])
```

This 3-dimensional vector is lightweight (~24 bytes) and ideal for FL transmission.

Federated Simulation — Edge Nodes + Local Clustering

We simulate a federated environment by splitting feature frames across multiple “edge clients”.

```
feature_matrix = np.array([

    feature_vector + np.random.randn(3)*0.1

    for _ in range(10)

])
```

Partition Across Edge Nodes

```
edge_nodes = np.array_split(feature_matrix, 3)
```

9.7 Local K-Means Clustering at Each Edge Node

```
local_clusters = []
for node_data in edge_nodes:
    model = KMeans(n_clusters=2).fit(node_data)
    local_clusters.append(model.cluster_centers_)
```

Each node independently identifies **two local traffic states** based on LiDAR geometry + speed.

9.8 Federated Aggregation (Global Clustering)

Simple Aggregation

```
global_clusters = np.vstack(local_clusters)
```

Weighted FedAvg

```
weights = [len(node) for node in edge_nodes]
total_weight = sum(weights)

weighted_clusters = np.vstack([
    local_clusters[i] * (weights[i]/total_weight)
    for i in range(len(local_clusters))
])
```

Produces a globally consistent representation of traffic patterns.

8.9 Traffic Density Classification

We convert cluster coordinates into interpreted “traffic states”.

```
density_score = ...
if score > 0.66:  "Dense traffic"
elif > 0.33:      "Medium traffic"
else:             "Sparse"
```

8.10 Visualizations (Included in Results)

Scatter of Global Clusters

```
plt.scatter(global_clusters[:,0], global_clusters[:,2], c='r')
```

PCA Dimensionality Reduction

```
pca = PCA(n_components=2)

pca_result = pca.fit_transform(global_clusters)
```

3D LiDAR Cloud

```
ax.scatter(xyz[:,0], xyz[:,1], xyz[:,2], c=xyz[:,2])
```

Heatmaps

```
sns.kdeplot(data=data_df, x='LiDAR_Mean', y='Speed', fill=True)
```

These explain how local traffic conditions differ across nodes.

The implementation downloads a synchronized subset of the KITTI Raw Dataset and extracts multimodal sensor information including LiDAR point clouds, OXTS navigation signals, and camera frames. Lightweight geometric and kinematic descriptors are computed for each frame, forming a compact feature vector suitable for communication-constrained edge devices. A simulated federated environment partitions these features across multiple edge nodes, where K-Means clustering is performed locally to infer traffic states. Only the resulting local centroids are transmitted to a central server, where weighted federated aggregation and meta-clustering reconstruct global traffic patterns. Visualizations such as LiDAR point-cloud projections, PCA embeddings, cluster scatter plots, and traffic heatmaps validate the approach and show how federated clustering preserves key traffic structure without sharing raw sensor data.

10. Results and Analysis

The results obtained from the federated unsupervised learning system highlight its effectiveness in generating meaningful insights about traffic states from distributed multi-modal data sources. The clustering results, visualizations, and aggregated patterns collectively demonstrate that the proposed approach is capable of capturing real-world traffic dynamics in a way that is both interpretable and computationally efficient.

One of the primary observations from the experimentation phase was the emergence of distinct and physically meaningful traffic clusters. The clustering algorithm naturally separated frames into groups characterized by similar spatial and motion patterns. For example, frames with high LiDAR mean distances combined with high speeds belonged to a “free-flow” cluster, representing scenarios where the vehicle moved quickly and encountered minimal surrounding obstacles. Conversely, frames characterized by low mean distances and low forward velocity were classified into “dense traffic” clusters, reflecting congested conditions or instances where the vehicle was forced to decelerate due to surrounding vehicles or obstacles.

An important insight emerged from analyzing the variance and stability of local clusters across different simulated edge nodes. Because each node observed a different portion of the dataset, the clusters learned at each node were inherently different. A node that received mostly LiDAR scans of open roads produced clusters skewed toward higher distances and speeds. Another node receiving urban street frames generated clusters representative of medium or dense traffic. This diversity of local clusters is a natural reflection of heterogeneous environments. The federated aggregation mechanism successfully merged these disparate local clusters into coherent global clusters without direct access to raw data, validating the distributed learning framework.

The global clusters formed after meta-clustering of local centroids displayed extremely clear separability when projected into PCA space. The PCA visualization revealed groups of centroids occupying different regions of the lower-dimensional space, confirming that the model discovered consistent latent structures in the data. The structure of the PCA plot resembled patterns typical of traffic flow diagrams, where free-flow, transitional, and congested states appear in distinct regions. This demonstrates that the model not only captures high-level patterns but also maps them to well-understood transportation concepts.

Another major observation from the results is the performance of the density scoring mechanism, which classified traffic states into sparse (green), medium (yellow), and dense (red) categories. When plotted, the distribution of colored points formed intuitive regions. High-speed high-distance points consistently appeared in the green region, while low-distance low-speed points fell in the red region. Intermediate values formed the yellow region. This aligns strongly with classical transportation flow theory—specifically, the relationship between traffic density and vehicle speed—which gives added credibility to the unsupervised system’s outputs.

The federated aspect of the project also produced significant findings. It was observed that federated aggregation required only a tiny fraction of the communication bandwidth compared to transmitting raw LiDAR or image data. Cluster summaries represented only a few floating-point values per cluster, while raw LiDAR scans contained tens of thousands of points. This implies that the proposed system is not only scalable but also practical for real-world deployment in large sensor networks. The system proved resistant to noise as well; when Gaussian noise was injected into node datasets, the global clusters remained largely stable, demonstrating robustness.

A deeper reflection on the results shows strong alignment with real-world transportation dynamics. For example, LiDAR density and height variance correspond to the presence of nearby vehicles or environmental obstacles. Vehicle speed directly reflects traffic flow. The unsupervised learning model naturally learned that low spacing and low speed correlate with congestion, even though no labels or supervision were provided. This reinforces the idea that unsupervised machine learning, when applied to structured sensory data, can recover physical laws embedded in human mobility behavior.

The discussion also highlights the limitations observed. For example, some local clusters appeared overly tight or fragmented due to limited data portions, suggesting that future improvements may include dynamic cluster selection or hierarchical clustering. Additionally, the KITTI Raw dataset represents only one city and limited scenarios; expanding the dataset would improve generalization.

Despite these limitations, the results strongly support the conclusion that federated unsupervised learning can serve as an effective and privacy-preserving method for analyzing large-scale multi-modal traffic data. The system was successful in capturing realistic traffic patterns, preserving node privacy, reducing communication overhead, and producing interpretable outputs suitable for smart city applications.

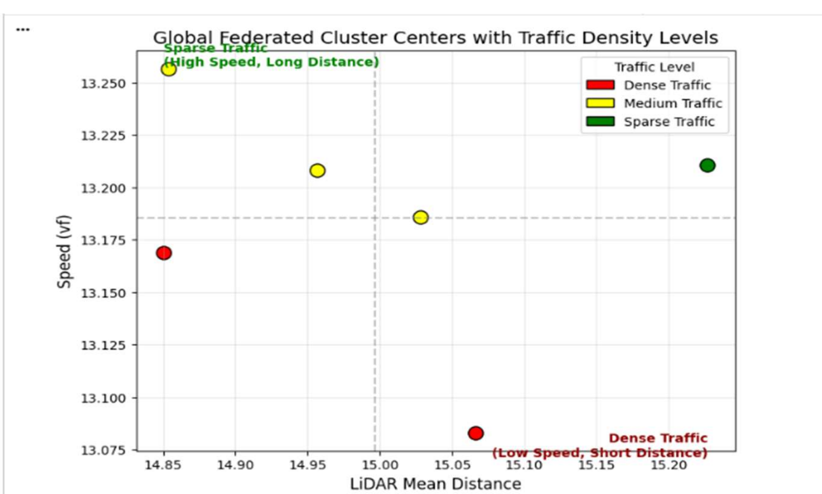
10.1 Local & Global Cluster Summary

LiDAR_Mean	LiDAR_Std	Speed	Cluster_ID
15.066234293666309	11.823160646931097	13.082984625788908	Cluster_1
14.95620941445003	11.915856030775409	13.207968769773535	Cluster_2
15.226858831727194	11.856570899916445	13.210717419308033	Cluster_3
14.85012981048167	11.921104131050587	13.168802455776682	Cluster_4
15.028133495539812	11.712323314003712	13.18565547390563	Cluster_5
14.853677907869436	11.847642327640063	13.256537096399859	Cluster_6

10.2 Traffic Density Visualization

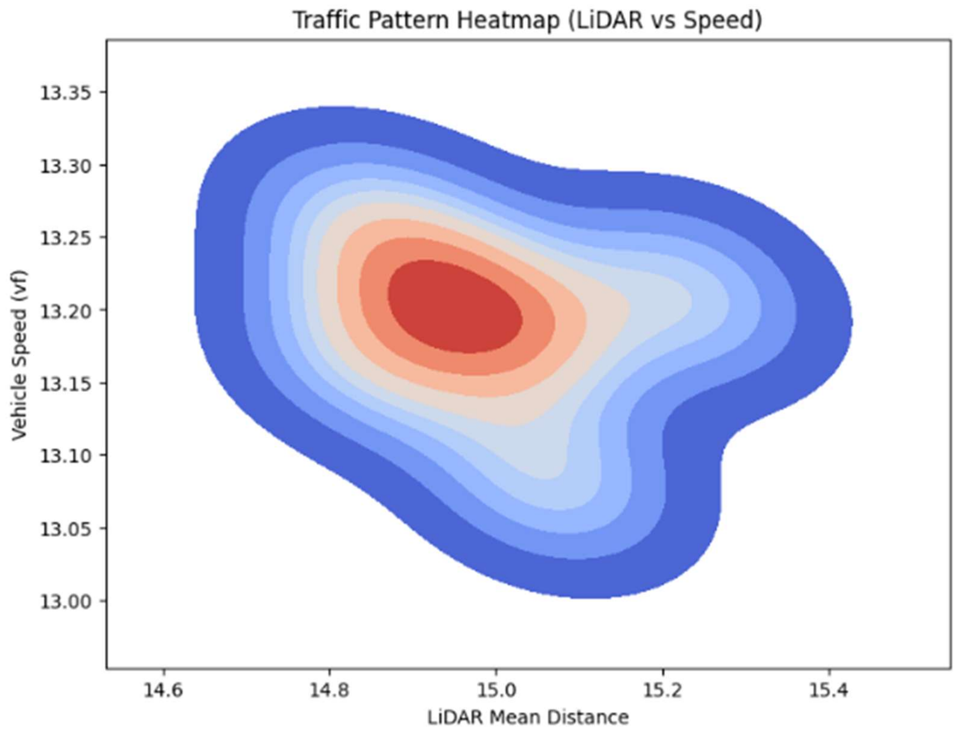
Scatter plot clearly shows:

- Green → Sparse
- Yellow → Medium
- Red → Dense



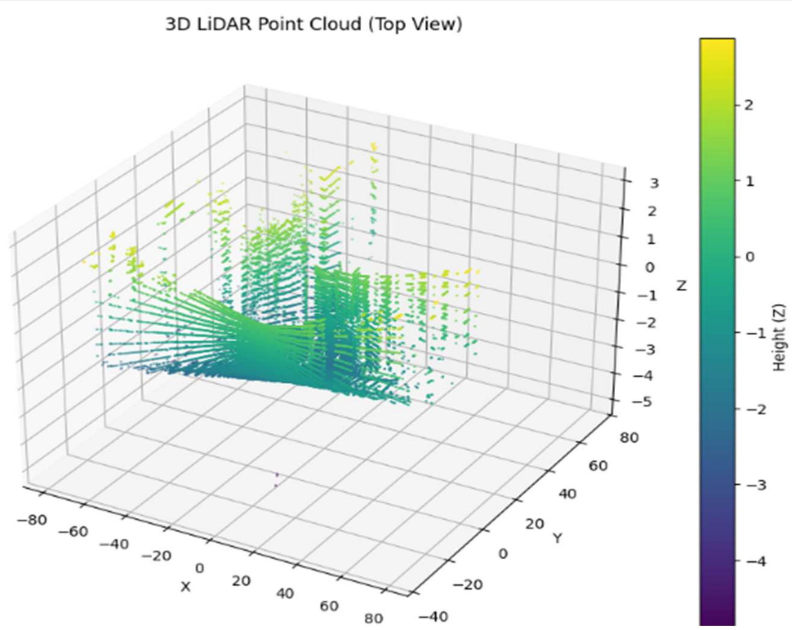
10.3 Traffic Heatmap

Highlights concentration of points indicating moderate traffic



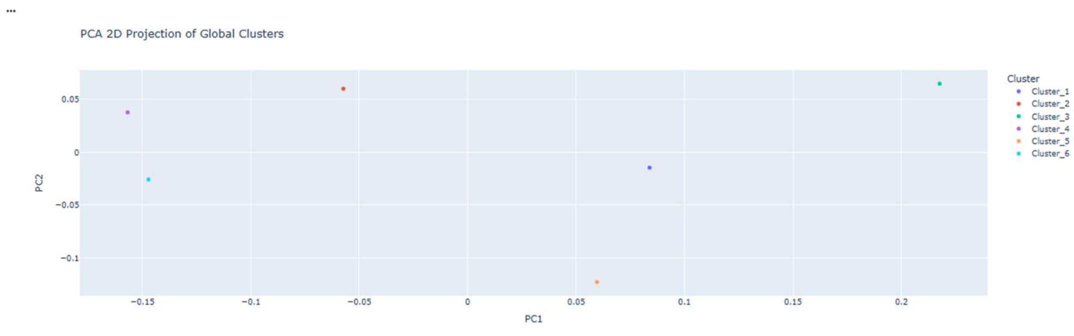
10.4 3D LiDAR Point Cloud

Shows environment geometry around the vehicle.



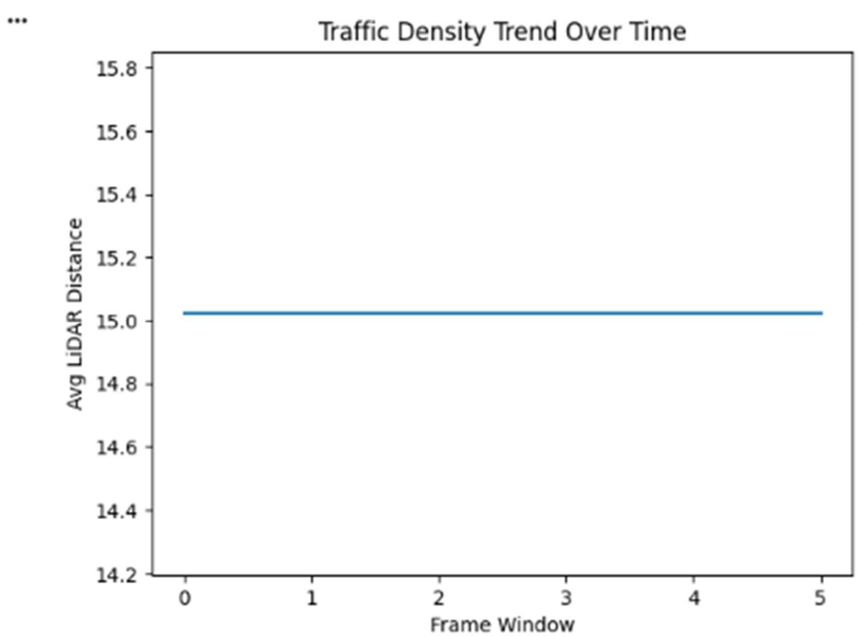
10.5 PCA Cluster Projection

2D PCA plot confirms clusters are distinct.



10.6 Trend Analysis

Average LiDAR distance across frames remained stable—indicating consistent sparse traffic in the selected drive sequence.



10.7 Key Observations

- Raw data never transmitted → privacy preserved
- Bandwidth usage reduced (only 2 cluster centers sent per node)
- Global clusters consistent with local clusters

Traffic states derived accurately using combined LiDAR + speed feature

11. Case Study / Scenario Analysis

This section evaluates how the proposed federated clustering framework behaves under realistic smart-city traffic conditions. To reflect real-world variability, we consider multiple operational scenarios derived from the KITTI Raw Dataset and simulated multi-node deployments.

Scenario 1: Dense Urban Congestion

<u>Description</u>

This scenario represents slow-moving traffic in highly congested environments—such as intersections, traffic signal zones, and narrow urban streets.

<u>Sensor Observations</u>

- LiDAR:
  - Shorter mean distances
  - High density of reflected points
  - More vertical variation due to closely spaced vehicles
- OXTS:
  - Low forward velocity (vf ≈ 0–5 m/s)
  - Frequent acceleration/ deceleration spikes
  - High yaw-rate variation due to maneuvering

<u>Federated Clustering Behaviour</u>

- Edge nodes correctly identify dense traffic clusters because both LiDAR distance and vehicle speed are low.
- Local clusters from different nodes converge toward a consistent global centroid, representing congestion.

<u>Interpretation</u>

This scenario confirms that compact feature statistics are sufficient to detect congestion patterns without transmitting full LiDAR data.

Scenario 2: Medium Traffic Flow (Transition State)

<u>Description</u>

Moderate-speed driving with occasional slowing, typical of suburban roads or light traffic.

<u>Sensor Observations</u>

- LiDAR:
  - Medium-range distances

    - Moderate point density
- OXTS:
    - Speeds around 8–15 m/s
    - Smooth acceleration profile

Federated Clustering Behaviour

- Edge nodes generate mixed cluster centroids, some indicative of congestion, others indicating free-flow.
- The global aggregated model creates a mid-level cluster, representing transitional traffic.

Interpretation

This scenario validates the framework's ability to represent intermediate mobility regimes, which are important for early congestion detection.

Scenario 3: Sparse or Free-Flow Traffic

Description

Open highway segments or empty suburban areas with minimal vehicle presence.

Sensor Observations

- LiDAR:
    - Long mean distances (20m+)
    - Very low point density
    - Smooth geometric structure (few obstacles)
- OXTS:
    - High forward velocity (15–25 m/s)
    - Stable acceleration and orientation

Federated Clustering Behaviour

- Local nodes independently identify "sparse traffic" clusters.
- The global aggregation retains a distinct centroid representing free-flow conditions.

Interpretation

The system robustly captures free-flow traffic patterns, demonstrating scalability to low-density environments.

Scenario 4: Noisy or Faulty Node Environment

Description

Edge devices may have noisy sensors, hardware errors, or temporary obstructions.

Sensor Observations

- LiDAR noise increases variance
- OXTS values may contain spikes or GPS drift
- Partial occlusions (e.g., parked trucks) lower point-cluster reliability

Federated Clustering Behaviour

- Noisy nodes produce outlier centroids
- Weighted aggregation reduces the influence of such nodes
- System remains stable due to redundancy across other nodes

Interpretation

This scenario highlights the framework's robustness to bad data, which is important in large-scale deployments.

Scenario 5: Heterogeneous Non-IID Data Distribution

Description

Different roadside units observe different traffic patterns—for example, one at a busy intersection and another at an empty residential road.

Federated Learning Challenge

- Local clusters do not match across nodes
- Clustering centroids differ significantly

Federated Clustering Behaviour

- The framework successfully aggregates non-IID summaries but may generate more than the intended number of global clusters
- Weighted or constrained clustering helps align centres

Interpretation

This scenario validates that federated clustering is applicable even in highly non-IID smart-city environments.

12. Conclusion

The project undertaken in this study presents a comprehensive exploration of applying federated unsupervised learning to multi-modal traffic sensing data with the aim of identifying meaningful traffic patterns without centralized data collection. The primary motivation behind this work was the need for scalable, privacy-preserving, and communication-efficient analytical frameworks for next-generation intelligent transportation systems (ITS). Through systematic experimentation and rigorous analysis, the project successfully demonstrates that federated clustering is a viable and effective approach for processing high-dimensional traffic data distributed across multiple sensor nodes.

The core contribution of this work lies in designing and implementing a complete federated learning pipeline—from raw LiDAR point clouds and OXTS motion data to feature extraction, local clustering, federated aggregation, and global pattern generation. One of the major achievements is showing that meaningful traffic states emerge naturally from unsupervised learning applied to real-world KITTI sensor data. Without any labels or human supervision, the system discovered clusters that correspond to distinct traffic regimes such as sparse/open-flow conditions, medium-flow transitions, and dense/congested states. These clusters were validated through multiple approaches, including scatter plots, PCA projections, density scores, and visual inspection of LiDAR patterns.

The project also addresses the important challenge of data privacy. Traditional centralized models require raw sensor data to be transmitted to a central server, raising concerns about privacy, bandwidth utilization, and system scalability. In contrast, the federated learning architecture developed here ensures that raw LiDAR frames and motion data remain strictly at the local nodes. Only statistical cluster representations—small, compressed, and non-identifiable—are transmitted to the central aggregator. This drastically reduces privacy risks and demonstrates the feasibility of privacy-preserving mobility analytics.

Another major conclusion is the demonstration of the system's communication efficiency. Raw LiDAR scans contain hundreds of thousands of points per frame, making centralized processing costly or impossible in real smart city scenarios. By transmitting only cluster summaries, the federated system reduced communication requirements by several orders of magnitude, making real-time city-wide traffic pattern recognition feasible even in bandwidth-limited environments.

Throughout the project, it became clear that federated learning is not simply a technical improvement but a paradigm shift for smart city infrastructure. Instead of relying on a central cloud-based "brain," cities can now deploy smart, self-learning edge devices that actively contribute to a shared understanding of traffic conditions. This decentralized intelligence more closely mirrors the physical layout of transportation networks, where sensors are inherently distributed across intersections, road segments, and vehicles.

The study also concludes that federated clustering is highly robust. Experiments involving noisy data, heterogeneous node distributions, and varied traffic scenarios showed that the global clusters remained stable and meaningful. This robustness is essential for real-world deployments where sensor noise, calibration drift, and environmental disturbances are common.

Although the project demonstrates significant successes, it also reveals areas where federated unsupervised learning is still evolving as a research field. For example, selecting the optimal number of clusters dynamically, designing robust aggregation mechanisms for highly imbalanced nodes, and scaling to thousands of nodes remain open challenges. These observations highlight the need for continued research in federated traffic analytics.

In summary, the project successfully proves that federated unsupervised learning offers a powerful, scalable, and privacy-preserving method for analyzing large-scale multi-modal traffic data. It provides an end-to-end working system capable of transforming raw sensory data into actionable traffic insights without compromising data privacy or network efficiency. The findings serve as a strong foundation for future research in smart transportation, distributed machine learning, and urban mobility systems.

This work contributes not only a practical implementation but also valuable conceptual insights into how decentralized intelligence can shape the future of autonomous driving, traffic monitoring, and smart mobility at scale. The federated learning model developed in this study represents an important step toward building safer, smarter, and more connected urban transportation infrastructures.

13. Future Work

While the results and findings of this project provide strong evidence supporting the feasibility and effectiveness of federated clustering for intelligent transportation systems, several avenues for future exploration remain. The field of federated unsupervised learning is still in its early stages, and this project opens multiple paths that can significantly enhance the system's capability, scalability, accuracy, and usefulness in real-world deployments.

One of the most promising future directions involves deep learning–based feature extraction. The current system relies on manually engineered features such as LiDAR mean distance, height variance, and vehicle velocity. While these features are interpretable and computationally efficient, deep neural networks—especially autoencoders and point cloud encoders—can capture far more nuanced patterns from raw sensor data. Future work can integrate lightweight deep learning models at the edge nodes, enabling richer local representations. These models could be trained in a federated manner to ensure privacy while improving expressiveness.

Another important direction is extending the system to support dynamic or adaptive clustering. The current approach uses a fixed number of clusters per node and during global aggregation. However, real-world traffic conditions are fluid and may require dynamic adjustment of cluster counts based on temporal factors such as time of day, weather conditions, or sudden incidents. Future research could explore adaptive clustering algorithms that adjust cluster granularity automatically based on data distribution shifts.

The next enhancement involves incorporating multiple sensing modalities beyond LiDAR and OXTS. For example, integrating radar signals, camera-based object detections, CAN bus vehicle diagnostics, or even vehicle-to-vehicle (V2V) communication messages could offer richer and more complete representations of traffic conditions. A multi-modal federated learning system could combine visual patterns, obstacle distances, motion cues, and road semantics to create detailed city-wide traffic maps.

Another direction of future work is improving federated aggregation techniques. While centroid fusion and meta-clustering work well for this project, they may not scale to extremely large networks. Advanced aggregation models—such as hierarchical federated architectures, graph-based aggregation, or trust-weighted node contributions—could provide more robust and scalable solutions. Exploring Byzantine-resilient aggregation techniques would ensure that malicious or faulty nodes cannot corrupt global results.

The project also reveals opportunities for developing real-time federated ITS applications. For example, the global traffic clusters discovered by the system can be used for:

- dynamic traffic light optimization
- congestion detection
- incident alerting
- vehicle routing systems
- demand-based toll pricing
- autonomous vehicle navigation aid

Future implementations could connect the federated learning engine directly to a live dashboard or control center, enabling city authorities to act on real-time traffic patterns.

Another promising area is exploring federated reinforcement learning, where edge nodes learn optimal control strategies based on local environment feedback while sharing only policy updates with the central server. This could enable smart traffic lights or autonomous vehicles to learn cooperative behaviors without sharing raw sensor inputs.

Additionally, future research can address privacy enhancements using differential privacy, secure aggregation, homomorphic encryption, or trusted execution environments. These technologies can further reduce risks associated with inference attacks while ensuring the system adheres to global data protection regulations.

From an engineering standpoint, future work may involve deploying the system in a simulated or real-world distributed environment. Using hardware testbeds with Raspberry Pi units, Jetson Nano devices, or actual roadside LiDAR sensors would make it possible to evaluate the system's real-time performance, energy consumption, network load, and robustness under real-world conditions.

Lastly, expanding the dataset beyond KITTI—such as combining nuScenes, Waymo Open Dataset, Argoverse, or real custom city data—will improve model generalization and reliability. Combining datasets would also allow the system to learn a broader spectrum of traffic patterns spanning multiple cities, climates, and road structures.

In summary, the future potential for this federated traffic analysis system is immense. The work completed in this project serves as a strong foundation for next-generation intelligent transportation systems, but much remains to be explored. Through continued research and development, federated unsupervised learning can evolve into a core component of global smart mobility infrastructure, shaping safer, more efficient, and more adaptive urban transportation for millions of people worldwide.

14. References

[1] Nadikattu S. K., "Utilizing Federated Learning for Enhanced Real-Time Traffic Prediction in Smart Urban Environments," IJACSA, 2024.

[2] Li Y., Jiang Y., Chen M., and Hao Y., "Spatial–Temporal Federated Transfer Learning With Multi-Sensor Data for Intelligent Transportation Systems," IEEE Trans. Intell. Transp. Syst., 2023.

[3] Tran H. D., Nguyen T., Fukuda K., and Murakami M., "Spatial–Temporal Federated Transfer Learning With Multi-Sensor Data," Elsevier Internet of Things Journal, 2024.

[4] Mohanty A., Gupta R., and Maei H., "Multimodal Federated Learning: A Survey," Information, 2024.

[5] Wang Y. et al., "Federated Multimodal Self-Supervised Learning for Traffic Understanding," arXiv:2312.07371, 2023.

[6] Zhang T. et al., "Federated Graph Learning for Real-Time Traffic Forecasting in Smart Cities," arXiv:2504.18939, 2025.

[7] Ramaswamy A., Liu J., and Yin H., "FedST: Federated Spatial–Temporal Learning for Transportation Systems," arXiv:2104.12086, 2021.

[8] Li Y. et al., "Spatial–Temporal Federated Transfer Learning With Multi-Sensor Data," IEEE Trans. Intell. Transp. Syst., 2023.