National College of Ireland

Adaptive Energy-Aware Federated Learning Framework for Traffic Flow Prediction in Smart Cities

MSc Research Project
MSc in Cloud Computing

Shwe Moe Thant
Student ID: 24170399

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shwe Moe Thant
Student ID:	24170399
Programme:	MSc in Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Aqeel Kazmi
Submission Due Date:	11/12/2025
Project Title:	Adaptive Energy-Aware Federated Learning Framework for Traffic Flow Prediction in Smart Cities
Word Count:	5199
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	26th January 2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Adaptive Energy-Aware Federated Learning Framework for Traffic Flow Prediction in Smart Cities

Shwe Moe Thant

24170399

Abstract

Traffic flow prediction increasingly adopts federated learning because it enables distributed model training without centralising sensitive mobility data. However, existing approaches still face challenges in communication overhead, device heterogeneity and limited energy awareness. Prior studies often focus on improving modelling accuracy through graph-based or meta-learning methods, or enhancing privacy through mechanisms such as differential privacy and encryption. Yet, they rarely examine variations in client energy consumption or adapt participation to real device conditions.

This study introduces the Adaptive Energy-Aware Federated Learning framework, which integrates a lightweight GRU predictor with client-side energy estimation, bandwidth-aware metadata and selective client participation. Differential privacy is incorporated to assess how privacy noise affects model performance and energy usage. The system uses containerised IoT clients and an S3-based coordination layer, and is evaluated on the Los-Loop, PeMSD8 and SZ-Taxi datasets against two baselines, FedAvg and FedProx.

Results show that AEFL maintains accuracy comparable to the baselines while reducing energy consumption in datasets with substantial client workload variation. Its behaviour converges to baseline performance when workloads are uniform. Differential privacy introduces modest changes in accuracy and energy at moderate noise levels. These findings demonstrate that adaptive and energy-conscious federated learning provides a practical and sustainable approach to traffic flow prediction in smart-city environments.

1 Introduction

Intelligent transportation systems are a defining component of modern smart cities, where real-time sensing and data-driven decision-making help manage congestion, improve routing and support large-scale mobility. Traffic flow prediction is central to this ecosystem because it enables early identification of speed fluctuations, congestion events and spatio-temporal disruptions. Traditional forecasting approaches rely on centralised deep learning pipelines that aggregate mobility data on a unified server for training. While these models demonstrate strong performance using graph neural networks and spatio-temporal architectures (Li et al., 2018), they require continuous transmission of large volumes of fine-grained mobility traces. This data movement raises concerns about privacy, communication overhead and long-term computational cost. These limitations intensify as

sensing infrastructures scale and cities deploy high-frequency monitoring from vehicles, roadside units and edge sensors.

Federated learning has emerged as a promising alternative for traffic prediction because it enables distributed model training without transferring raw data to a central server. Devices collaborate on model updates while keeping sensitive information such as GPS traces, speed records and individual mobility patterns local. Several studies demonstrate the potential of FL for spatio-temporal forecasting, including asynchronous graph convolutional models (Qi et al., 2023), graph representation-driven learning (Yuan et al., 2023) and concept-enhanced FL architectures (Wu et al., 2025). These systems highlight the viability of decentralised training for traffic prediction. However, despite improvements in model accuracy and privacy protection, practical deployment in smart cities remains challenging.

A key obstacle is the communication overhead associated with multi-round model exchange. Even when privacy-preserving techniques such as differential privacy or homomorphic encryption are applied (Al-Huthaifi et al., 2024; Deng and Shen, 2022), model transmissions remain costly and can overwhelm bandwidth-limited or intermittently connected devices. Another challenge is device heterogeneity. IoT clients in real traffic systems differ in computing power, battery capacity, mobility patterns and network availability. Research on meta-learning and adaptive federated frameworks addresses some of this variation (Sepasgozar and Pierre, 2022; Feng et al., 2024; Johnson and Geller, 2025), but energy efficiency often remains a secondary consideration. Recent sustainability-focused FL studies (Domini et al., 2025; Khatua et al., 2024) also note that energy-aware optimisation is rarely integrated into traffic prediction models, despite its importance for long-term deployment on resource-constrained nodes.

Energy awareness is especially relevant in smart-city contexts, where many sensing devices operate on limited or fluctuating power budgets. Battery-powered roadside units, mobile vehicles and independent sensor grids cannot sustain prolonged or intensive model exchange without risking premature dropout. Without mechanisms to adapt to energy fluctuations and bandwidth constraints, federated learning systems may experience stalled training, reduced convergence stability or biased model updates. Moreover, the interaction between privacy mechanisms, communication load and energy consumption remains poorly understood, leaving open questions on how to design systems that balance predictive performance, privacy and sustainability.

These challenges motivate the development of an adaptive federated learning framework that explicitly incorporates energy awareness and selective participation into the training process. This study examines how an FL system can dynamically respond to real-world device conditions while maintaining competitive accuracy for traffic flow forecasting. The framework also integrates differential privacy to evaluate how privacy protection interacts with energy cost and model performance. By addressing these interdependent requirements, this study aims to contribute a practical and sustainable learning paradigm for intelligent transportation systems.

1.1 Research Objectives

This study seeks to answer the following research question:
How can an adaptive federated learning framework be designed to jointly optimise energy efficiency, predictive accuracy and privacy protection on resource-constrained edge devices for real-time traffic flow prediction in smart

cities?

To address this question, the study pursues the following objectives:

- Develop an Adaptive Energy-Aware Federated Learning framework that regulates client participation according to real-time energy and bandwidth metadata.
- Employ a lightweight recurrent model suitable for resource-constrained devices while maintaining robust forecasting performance.
- Integrate model update compression and systematic logging to enable analysis of communication behaviour and its relationship to client energy consumption.
- Incorporate a differential privacy mechanism to examine the trade-off between privacy guarantees and model performance under adaptive scheduling.
- Evaluate the framework across three real-world datasets to assess its generalisability in diverse and heterogeneous traffic environments.

The remainder of this report is structured as follows. Section 2 presents a critical literature review of federated learning approaches for traffic prediction and identifies key limitations that motivate this study. Section 3 outlines the methodological design, including dataset preparation, experimental setup and evaluation procedures. Section 4 details the architectural design of the Adaptive Energy-Aware Federated Learning framework. Section 5 describes the practical implementation, including algorithms, modules and system components. Section 6 presents experimental findings with an integrated analysis of accuracy, energy consumption and privacy behaviour. Finally, Section 7 concludes the study by summarising contributions, discussing limitations and outlining directions for future research.

2 Literature Review

Federated learning (FL) has become an important technique in intelligent transportation systems (ITS) because it enables traffic flow prediction (TFP) without the risks of centralised data aggregation. Existing research spans four main themes: graph-based spatio-temporal modelling, adaptive or meta-learning-based FL, privacy-preserving mechanisms, and early energy-aware optimisation. These strands contribute to distributed traffic forecasting, although progress remains uneven and often disconnected. This review summarises key developments within these themes and highlights how they inform the design of the Adaptive Energy-Aware Federated Learning (AEFL) framework proposed in this study.

2.1 Graph-Based Federated Learning for Spatio-Temporal Modelling

Spatio-temporal modelling in FL frequently adopts graph neural networks (GNNs), motivated by complex correlations in road networks. Studies show that GNNs capture spatial structure effectively and improve accuracy on benchmark datasets such as PeMS and SZ Taxi. FedAGCN (Qi et al., 2023), FedSTN (Yuan et al., 2023) and FC-FedGCN (Wu et al., 2025) demonstrate how attention modules, semantic encoders and asynchronous

aggregation enhance representation learning. However, these methods introduce substantial computational cost, relying on heavy graph convolutions and additional preprocessing that are not well suited to resource-constrained IoT clients. Moreover, few studies measure how these modelling choices influence energy consumption. Collectively, these approaches highlight a tension between representation power and practical deployability, motivating the lightweight GRU-based modelling strategy used in AEFL.

2.2 Meta-Learning and Adaptive Federated Frameworks

Meta-learning has been explored to address non-IID traffic patterns. Existing methods enhance FL adaptability by improving generalisation under dynamic mobility conditions. Approaches such as Fed-NTP (Sepasgozar and Pierre, 2022), DFML (Feng et al., 2024) and meta-learning-based FL (Johnson and Geller, 2025) show that recurrent encoders, meta-gradients and model-agnostic adaptation can reduce performance degradation when client distributions evolve. While these methods improve predictive stability, they also increase client-side computation and communication. Repeated adaptation raises energy consumption, and no study regulates participation based on client state or bandwidth quality. Thus, existing adaptive FL methods improve robustness but overlook energy implications, motivating AEFL's client selection strategy based on energy and bandwidth metadata.

2.3 Privacy-Preserving and Secure Federated Architectures

Given the sensitivity of mobility data, many FL frameworks incorporate privacy-preserving mechanisms. Studies such as FedGODE (Al-Huthaifi et al., 2024), PPSTSL (Feng and Qian, 2025) and HE-enabled LSTM models for vehicular networks (Deng and Shen, 2022) show that differential privacy, homomorphic encryption and split learning strengthen privacy guarantees. These protections typically increase computational and communication overhead, creating practical barriers for large-scale ITS deployment. In addition, most privacy-focused studies evaluate privacy and accuracy independently, without examining effects on resource consumption. Across the literature, privacy-enhancing techniques are often treated as isolated modules rather than interacting with energy and adaptivity. AEFL addresses this gap by combining local DP mechanisms with explicit energy estimation, enabling direct analysis of privacy–accuracy–energy trade-offs.

2.4 Energy-Aware and Sustainable Federated Learning

Compared to modelling and privacy research, energy-aware FL for ITS remains underdeveloped. Existing studies recognise sustainability concerns but rarely include explicit energy measurement. Techniques such as sparsification and quantisation, explored by SParSeFuL (Domini et al., 2025), reduce transmission and computation, although aggressive pruning can harm convergence stability. Other work examines system-level optimisation, such as genetic optimisation of FL-enabled Internet of Vehicles routing (Khatua et al., 2024), but avoids detailed characterisation of training energy. Related studies such as Fed-CI (Zhang et al., 2025) suggest reduced power use through personalised updates, yet do not quantify energy consumption. Overall, progress remains fragmented, with no unified framework integrating lightweight modelling, adaptive participation and direct energy measurement. AEFL addresses this gap by combining these

elements in a single implementation designed for heterogeneous IoT environments.

2.5 Comparative Assessment of Federated Learning Approaches for TFP

Table 1 summarises representative FL approaches for traffic flow prediction across five dimensions: spatio-temporal modelling, privacy mechanisms, adaptivity, energy efficiency and real-world evaluation. The overview shows that many studies provide strong modelling and privacy support, while energy-aware participation and systematic energy analysis remain limited. Existing frameworks either assume energy benefits or infer them indirectly. None combine all required elements in a single empirically validated architecture. AEFL builds on these observations by integrating lightweight GRU modelling, differential privacy, adaptive client selection and explicit energy accounting, enabling a more complete understanding of the trade-offs involved.

Table 1: Comparative overview of representative FL approaches for traffic flow prediction across modelling, privacy, adaptivity, energy and real-world evaluation dimensions.

Paper / Study	**Spatio-Temporal**	**Privacy Mechanism**	**Adaptivity / Meta-Learning**	**Energy Efficiency**	**Real-World Evaluation**
Qi et al. (2023) FedAGCN	●	◐	○	○	◐
Yuan et al. (2023) FedSTN	●	●	○	○	◐
Wu et al. (2025) FC–FedGCN	●	◐	○	○	◐
Sepasgozar and Pierre (2022) Fed–NTP	●	◐	◐	○	○
Feng et al. (2024) DFML	●	◐	●	○	○
Johnson and Geller (2025) Meta–FL	●	○	●	○	◐
Al-Huthaifi et al. (2024) FedGODE	●	●	○	○	◐
Feng and Qian (2025) PPSTSL	●	●	○	◐	●
Deng and Shen (2022) FL–LSTM VANET	●	●	○	○	○
Domini et al. (2025) SParSeFuL	◐	◐	○	●	●
Khatua et al. (2024) IoVT FL	◐	◐	○	◐	●
Zhang et al. (2025) Fed-CI	●	◐	◐	◐	◐
Proposed AEFL Framework	●	●	●	●	●

Legend: ● = Fully addressed; ◐ = Partially addressed; ○ = Not addressed

Overall, the literature shows strong interest in improving FL for ITS, while energy behaviour, adaptive scheduling and privacy are often examined independently. Table 1 highlights the imbalance between modelling advances and sustainability considerations. AEFL addresses this gap through an integrated design that supports lightweight computation, adaptive participation and explicit energy measurement, providing a comprehensive foundation for large-scale, privacy-preserving TFP in heterogeneous environments.

3 Methodology

This study adopts a quantitative, experiment-driven methodology to design, implement and evaluate the Adaptive Energy-Aware Federated Learning (AEFL) framework for

traffic flow prediction in intelligent transportation systems. The methodological structure reflects federated learning characteristics, edge-device constraints and the need for systematic comparison across datasets, learning modes and privacy configurations.

3.1 Research Approach

The research follows a structured experimental workflow focused on building an operational federated learning system and observing its behaviour under controlled conditions. A quantitative approach is appropriate because the objectives involve measurable outcomes, including energy consumption, communication volume, predictive accuracy and the impact of differential privacy noise. These outcomes arise from model execution and client-device interactions, making experiments suitable for reliable assessment.

The workflow proceeds in three stages. First, benchmark datasets are preprocessed and partitioned into heterogeneous client segments to emulate non-IID characteristics of smart-city sensing systems. Second, the AEFL framework is implemented, incorporating lightweight GRU-based forecasting, adaptive client scheduling, energy estimation and optional privacy and compression modules. Finally, controlled experiments are conducted across multiple datasets to assess how adaptivity influences performance compared with established federated learning baselines. All models follow identical training configurations to ensure fair comparison, and the resulting metrics provide a basis for analysing how energy-aware decision-making affects federated learning behaviour.

3.2 Data Collection and Preprocessing

Three benchmark datasets were used to evaluate the framework: PeMSD8 (Chen et al., 2021), SZ-Taxi (Zheng et al., 2011) and Los-Loop (Jagadish et al., 2014). These datasets were selected because they differ in spatial density, traffic patterns and temporal variability, enabling assessment across diverse mobility environments.

Table 2: Overview of Benchmark Datasets

Dataset	**Sensors/Nodes**	**Edges**	**Time Steps**	**Time Range**	**Location**
PeMSD8	170	295	17,856	Jul–Aug 2016	California District 8
SZ-Taxi	156 zones	162	33,408	Jan–Feb 2015	Shenzhen, China
Los-Loop	201	250	34,272	Mar–Jun 2012	Los Angeles, CA

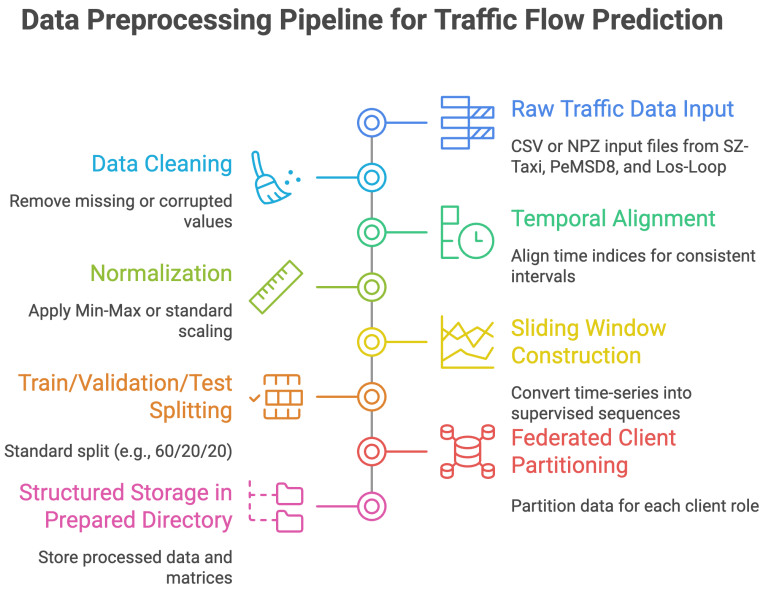

Figure 1: Data preprocessing pipeline for traffic flow prediction.

Each dataset undergoes a unified preprocessing pipeline comprising missing value imputation, temporal alignment, min–max normalisation and sliding window construction for supervised learning. Data splits follow a standard 70% training, 10% validation and 20% test configuration. Processed datasets are partitioned across simulated IoT roles, such as roadside units, vehicles, buses, cameras and regional sensors, generating distinct non-IID distributions. All outputs are stored in a structured directory layout to ensure reproducibility and support automated data loading during experiments.

3.3 System Model and Experimental Design

The experimental system integrates federated computation, container-based client execution and a central coordination server. Although deployment occurs on a single computational host, the design reflects cloud-edge federated learning platforms found in real smart-city environments. This alignment ensures that measured communication behaviour, client scheduling and energy effects represent realistic patterns rather than idealised simulations.

The system consists of a federated server that manages client selection, global aggregation and evaluation routines, and a set of Docker-based clients that emulate IoT devices. Each client holds a separate data partition and performs local training on a lightweight GRU model tailored to embedded device constraints. Assigning each client a role with distinct data scale and characteristics captures practical variability in workload and communication cost.

The experimental design examines how adaptivity, energy awareness and privacy mechanisms shape federated learning performance. Hyperparameters, learning modes and noise levels are defined through configuration files to ensure comparability. Energy consumption is estimated using logged measurements of computation time and communication volume. Although this method approximates device behaviour rather than measuring it directly, it provides a consistent basis for comparing AEFL with baseline methods across datasets.

All models share identical learning rates, optimiser settings, window sizes and initialisation seeds. This controlled setup isolates the effects of adaptive scheduling and ensures that performance differences arise from learning modes rather than unrelated configuration changes. Logging infrastructure captures accuracy metrics, communication patterns, update sizes and energy values across rounds, enabling detailed examination of learning dynamics.

3.4 Evaluation Metrics

Energy efficiency is assessed by summing estimated computational and communication energy across all participating clients and rounds. Communication behaviour is evaluated by analysing the sizes of downloaded and uploaded model states. Prediction performance is measured using mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE), which provide complementary views of prediction accuracy, sensitivity to large deviations and relative error.

Privacy-related effects are evaluated by examining accuracy degradation and energy variation under increasing levels of differential privacy noise. The study focuses on how noise affects federated learning stability rather than providing formal privacy guarantees, which would require deeper analysis beyond the scope of the implemented system.

3.5 Reproducibility and Validity

Reproducibility is maintained through deterministic random seeds, consistent data splits and standardised preprocessing pipelines. Containerised clients ensure stable execution across repeated runs and avoid inconsistencies arising from dependency differences. Automated scripts enforce uniform hyperparameters, mode selection and dataset handling across experiments, providing a reliable basis for comparison.

Validity is strengthened through widely recognised benchmark datasets, transparent reporting of preprocessing steps and established baseline algorithms. While energy estimation relies on software-based measurements rather than hardware power sensing, the approach remains suitable for comparative analysis because it is applied consistently across all methods and datasets. Limitations of this approximation are acknowledged in the discussion chapter, along with directions for improving measurement fidelity in future work.

4 Design and Implementation

The Adaptive Energy-Aware Federated Learning (AEFL) framework is designed to reproduce realistic smart-city traffic prediction conditions while remaining analytically transparent. Its architecture reflects constraints commonly observed in heterogeneous edge environments, including varied computational resources, fluctuating bandwidth and strong privacy requirements. This section outlines the architectural design, system workflow and supporting subsystems that underpin the detailed implementation presented in Section 5.

4.1 Architectural Overview

The framework adopts a distributed architecture in which a central federated server coordinates a set of containerised IoT clients. Each client emulates a distinct sensing role,

including roadside units, vehicles, buses, cameras and lightweight sensors. Communication occurs through an S3-style object storage layer that supports deterministic and decoupled exchange of model artefacts. The architecture is implemented in Python using PyTorch for model training, Docker for client isolation and Boto3 for storage communication. Together, these components create an extensible environment supporting adaptive scheduling, differential privacy, compression techniques and detailed energy profiling.

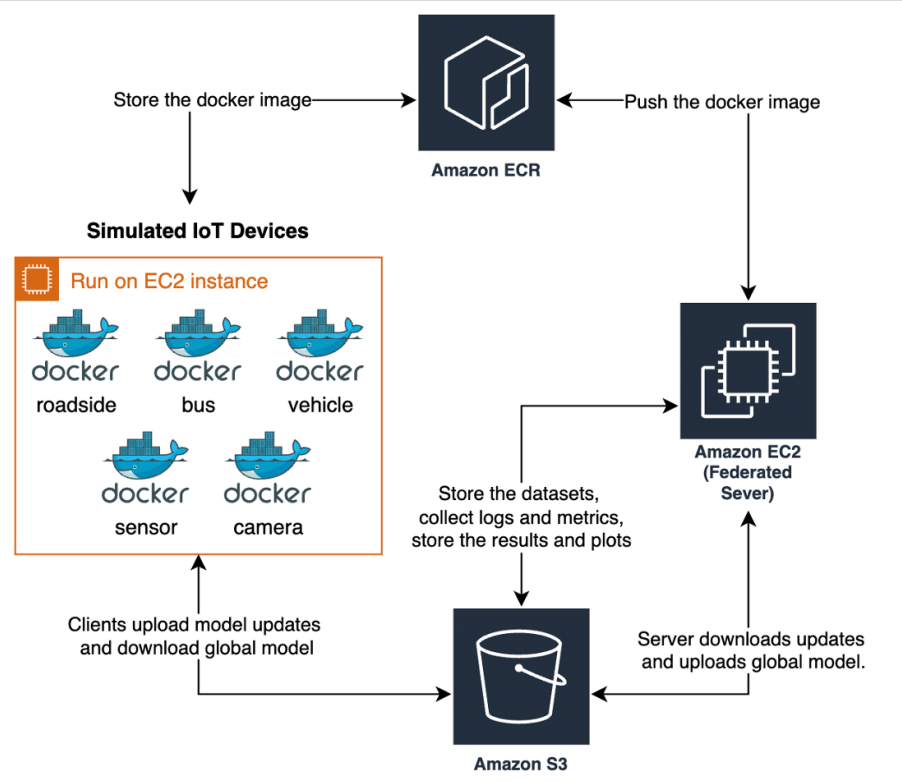

Figure 2: Overall architecture of the Adaptive Energy-Aware Federated Learning (AEFL) framework. Source: Author-generated.

4.1.1 Federated Server

The federated server orchestrates the training lifecycle. It initialises the global model, distributes parameters at the beginning of each round, collects client updates and performs aggregation according to the selected learning mode. FedAvg (McMahan et al., 2023) and FedProx (Li et al., 2020) apply uniform aggregation, while AEFL performs weighted aggregation based on bandwidth and energy conditions observed in the preceding round. The server stores global checkpoints, performs validation and test evaluation and generates summary metrics and energy reports. Server-side events, including upload and download latencies, update sizes, aggregation times and per-round metadata, are logged in structured JSON records to support reproducibility.

4.1.2 Containerised IoT Clients

Clients are instantiated as independent Docker containers, each holding a non-IID dataset partition and communicating exclusively through the S3 backend. During each round, a client retrieves the latest global model, performs local training on its assigned data slice

and uploads the resulting update. Each client also computes an energy estimate based on recorded training duration, communication volume and an associated power constant. Modules apply differential privacy or compression when enabled. This container-based design isolates execution across roles, supports repeatability and enables consistent evaluation across simulations.

4.1.3 S3-Based Coordination Mechanism

All communication between the server and clients is mediated through S3. Clients poll the storage layer for the latest global checkpoint and upload updates and metadata after local training. The server monitors the storage prefix to detect round completion and publish the next checkpoint. This mechanism mirrors cloud-based federated learning deployments in which devices rely on a storage-backed coordination layer rather than direct server communication. The structured organisation of objects also enables the framework to record transfer latency and communication volume for later analysis.

4.1.4 Energy, Privacy and Compression Modules

Three auxiliary subsystems enhance framework practicality. The energy module estimates computation and communication energy per client per round and feeds these values into AEFL's adaptive selection mechanism. The privacy module supports Gaussian differential privacy, enabling controlled examination of privacy–accuracy trade-offs. The compression module provides sparsification, top-k selection and 8-bit quantisation. All modules operate independently of core training logic and can be selectively enabled through configuration flags.

4.2 System Workflow

The system follows a structured workflow reflecting key stages of a federated learning cycle. At initialisation, the server loads the processed dataset, infers the number of graph nodes and constructs a lightweight GRU model. Residual artefacts from previous experiments are removed. Once the first global checkpoint is published, client containers begin listening for updates and load their respective data partitions.

Training proceeds over a fixed number of rounds. In each round, the server distributes the current global model, selected clients perform local updates, metadata is uploaded and the server aggregates received updates to form the next global checkpoint. AEFL uses recorded metadata to adjust future participation, prioritising clients with favourable energy or bandwidth characteristics. After completing all rounds, the server evaluates the final model on the test set and generates consolidated accuracy, energy and privacy summaries.

4.3 Design Considerations

The design of AEFL is guided by practical and methodological requirements. Energy awareness is central because smart-city devices exhibit varying power availability and communication cost. The framework records detailed metadata to support adaptive scheduling decisions. Modularity is achieved by implementing each subsystem as a dedicated Python module, simplifying debugging and extension. Reproducibility is maintained through deterministic random seeds, consistent directory structures and Docker-based

containerisation. Scalability is supported by introducing additional clients through new containers without modifying underlying coordination protocols.

4.4 Model Architecture

All experiments use a lightweight GRU-based model that balances predictive capacity with computational efficiency. The model processes temporal sequences through a GRU and maps the final hidden state to prediction values through a linear layer. This architecture is suitable for constrained IoT clients and supports controlled evaluation of federated mechanisms without confounding effects from more complex network structures.

5 Implementation

Building on the architectural design in Section 4, this section describes the practical implementation of the Adaptive Energy-Aware Federated Learning (AEFL) framework. The implementation integrates data processing, client simulation, adaptive scheduling, differential privacy, compression and server-side aggregation. Rather than presenting extensive pseudocode, this section focuses on technical design choices and component interactions that realise the system.

5.1 Software Stack and System Structure

The framework is implemented in Python on Ubuntu. PyTorch provides deep learning functionality, and NumPy handles preprocessing routines. Docker simulates heterogeneous IoT clients as isolated containers. An S3-compatible storage layer (via Boto3) enables decoupled communication between clients and the server. All runs generate structured JSON logs containing training durations, transfer sizes, metadata and energy estimates.

Table 3 summarises the main modules used in the implementation.

Table 3: Core modules and their implementation responsibilities.

Module	**Responsibilities**
`src/models/`	GRU architecture, forward pass, parameter initialisation
`src/fl/client/`	Local training loop, differential privacy, compression, metadata logging
`src/fl/server/`	Client selection, aggregation, global model management, evaluation
`data/`	Preprocessing, partitioning and dataset loading
`outputs/`	Accuracy metrics, plots, logs and energy profiles

5.2 IoT Client Implementation

Each client runs in a separate Docker container with a local dataset partition. At the start of each round, the client downloads the global model, performs several epochs of local training and uploads model updates with metadata.

5.2.1 GRU Model Computation

Clients use a lightweight GRU architecture suitable for resource-constrained devices. The GRU follows standard update rules:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t,$$

with conventional reset and update gates. This choice balances model capacity and computational simplicity.

5.2.2 Local Training Procedure

Local training is implemented as a mini-batch optimisation loop. The loss function is mean squared error, and FedProx introduces an additional proximity term when enabled. Clients use Adam with fixed hyperparameters across all modes and datasets. After training, updated parameter tensors are extracted and passed through optional differential privacy and compression transformations.

5.2.3 Differential Privacy Noise

Differential privacy applies Gaussian noise once per round:

$$\tilde{\theta} = \theta' + \mathcal{N}(0, \sigma^2 I),$$

implemented as element-wise tensor perturbation before upload.

5.2.4 Compression Methods

Three compression techniques are implemented:

- **Magnitude pruning**: parameters below threshold τ are zeroed.
- **Top-k selection**: retains a fraction α of the largest values.
- **8-bit quantisation**: parameters are scaled by the maximum absolute value and rounded to 8-bit integers.

These operations reduce upload size but are not evaluated extensively in Section 6, as discussed in the conclusion.

5.2.5 Energy Estimation

Energy consumption per round is computed using:

$$E = P_{\text{device}} T_{\text{train}} + c_{\text{net}}(D_{\downarrow} + D_{\uparrow}),$$

where T_{train} is measured on the client and D denotes transfer sizes recorded through the storage backend. This provides a consistent comparative measure across modes and datasets.

Table 4: Client roles and their dataset partition sizes.

Role	Nodes Assigned	Train Samples	Characteristics
Roadside	Few nodes	Small batch count	Low power
Vehicle	Medium	Moderate samples	Mobility-driven variability
Sensor	Few nodes	Small-medium	Stable temporal patterns
Camera	Many nodes	Larger volume	Higher power availability
Bus	Medium-large	Moderate-high	Mixed mobility characteristics

5.3 Client Roles and Dataset Partitions

Each client is assigned an IoT role, such as roadside sensor, vehicle or camera. Roles differ in dataset size, mobility characteristics and computational intensity. Table 4 summarises partition characteristics.

5.4 Federated Server Implementation

The server coordinates clients, manages system state and composes the global model.

5.4.1 Global Model Initialisation

For each dataset, the server infers the number of nodes and initialises a SimpleGRU model:

$$\theta_1 = \mathrm{GRUInit}(N, H).$$

Initial parameters are uploaded as the first global checkpoint.

5.4.2 Aggregation

Aggregation depends on the selected mode. FedAvg applies an unweighted mean, while AEFL uses metadata-derived weights:

$$\theta_{t+1} = \sum_{i \in \mathcal{S}_t} w_i \theta_i, \qquad w_i = \frac{s_i}{\sum_j s_j},$$

where s_i reflects each client's normalised bandwidth and energy profile.

5.4.3 AEFL Client Selection

At each round, the server scores clients using metadata from the previous round:

$$\mathrm{score}_i = 0.6 \left(\frac{b_i}{\max_j b_j} \right) + 0.4 \left(1 - \frac{e_i}{\max_j e_j} \right),$$

prioritising clients with high bandwidth and low energy consumption while filtering excessively costly participants.

5.5 Federated Learning Workflow

The training workflow consists of:

1. Initialising the global model and dataset partitions.
2. Selecting a subset of clients according to the chosen mode.
3. Distributing the global model to selected clients.
4. Running local training in each client container.
5. Applying optional differential privacy and compression.
6. Uploading model updates and metadata.
7. Aggregating updates and evaluating validation performance.

This process repeats for the configured number of rounds, with final evaluation performed on the test set.

5.6 Differential Privacy and Compression Modules

Table 5 summarises optional modules used in the experiments.

Table 5: Privacy and compression modules in the implementation.

Module	**Description**
Differential Privacy	Gaussian perturbation applied once per round
Magnitude Pruning	Removes parameters below magnitude threshold τ
Top-k Selection	Retains largest $k = \alpha \cdot \lvert p \rvert$ parameters
8-bit Quantisation	Symmetric scaling and integer rounding scheme

5.7 Experiment Automation

Automated Bash scripts control dataset selection, learning modes and differential privacy settings. Python scripts generate plots, summaries and CSV files under `outputs/`. This ensures the pipeline can be reproduced consistently across all datasets.

6 Evaluation and Results

This section evaluates the Adaptive Energy-Aware Federated Learning (AEFL) framework across three dimensions: energy consumption, predictive accuracy and behaviour under differential privacy noise. Experiments were conducted on the Los-Loop, PeMSD8 and SZ-Taxi datasets, enabling comparison across different spatial structures, traffic regimes and temporal patterns. Results are interpreted using descriptive comparisons rather than formal statistical tests, focusing on how dataset characteristics and adaptive scheduling shape overall behaviour.

6.1 Energy Consumption Results

Energy values were computed using the client profiling mechanism implemented in the framework. For each dataset, total energy consumption was aggregated across all clients and training rounds. Table 6 and Figure 3 summarise the results.

Table 6: Total Energy Consumption Across Methods and Datasets

Dataset	Mode	Total Energy (J)
SZ	AEFL	272.87
	FedAvg	307.94
	FedProx	299.31
PeMSD8	AEFL	2782.63
	FedAvg	2662.82
	FedProx	2745.48
Los	AEFL	236.26
	FedAvg	271.82
	FedProx	226.85

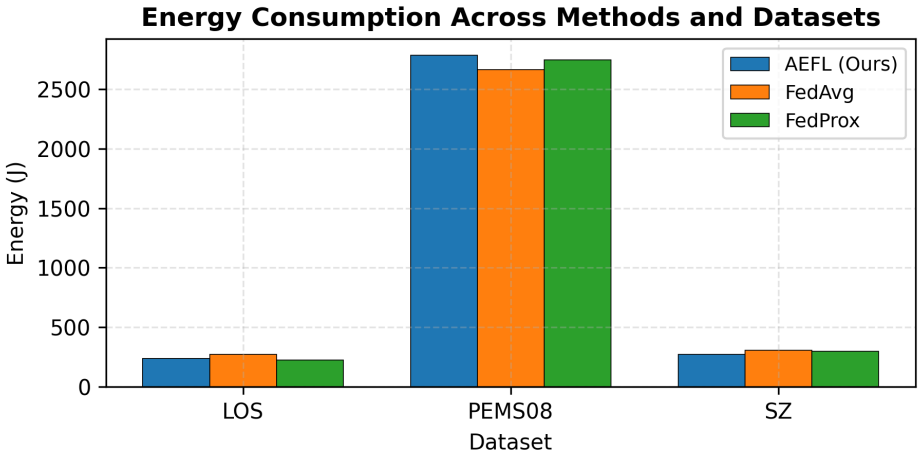

Figure 3: Energy consumption across methods and datasets.

6.1.1 Interpretation of Energy Outcomes

Differences reflect the heterogeneity and workload dynamics of each dataset. In SZ-Taxi, AEFL reduces total energy by around 11% relative to FedAvg. The dataset exhibits irregular taxi flows with sharp variation in local sample density over time. Clients associated with busy regions repeatedly incur higher computation and communication loads. By reducing participation from these roles, AEFL lowers overall energy use. This effect becomes more pronounced in later rounds, where AEFL suppresses clients whose metadata consistently indicate unfavourable cost–benefit profiles.

In Los-Loop, workloads are nearly uniform. Clients experience stable and predictable patterns, leaving AEFL with limited signals to differentiate roles. The small numerical differences fall within expected operational noise, illustrating a case where adaptivity provides limited benefit. This dataset represents a best-case scenario for baseline methods, and AEFL naturally converges to similar behaviour.

In PeMSD8, AEFL consumes slightly more energy than FedAvg but remains close to FedProx. The dataset shows intermediate variability, which can increase participation by clients that appear temporarily favourable. Short-lived bandwidth improvements or reduced communication latency can cause AEFL to admit more expensive clients. These fluctuations compound across rounds, producing minor overhead. However, the increase remains within the expected noise envelope of a moderately heterogeneous system.

Across datasets, these findings support **Objective 1**: AEFL reduces energy consumption when heterogeneity is strong while avoiding excessive overhead in more stable environments. The results also indicate that adaptivity does not introduce erratic energy fluctuations, which is important for deployments on battery-powered devices.

6.2 Predictive Accuracy Results

Prediction accuracy was evaluated using MAE, RMSE and MAPE. Table 7 and Figure 4 present the results.

Table 7: Prediction Accuracy Across Methods and Datasets

Dataset	Mode	MAE	RMSE	MAPE
SZ	FedAvg	0.5423	0.6968	1.5580
	FedProx	0.5524	0.7090	2.1544
	AEFL	0.5419	0.6943	1.8336
PeMSD8	FedAvg	0.7641	0.9342	1.3409
	FedProx	0.7850	0.9703	1.7986
	AEFL	0.7435	0.9181	1.2047
Los	FedAvg	0.01975	0.02278	0.8837
	FedProx	0.02117	0.02533	0.9466
	AEFL	0.01946	0.02279	0.8692

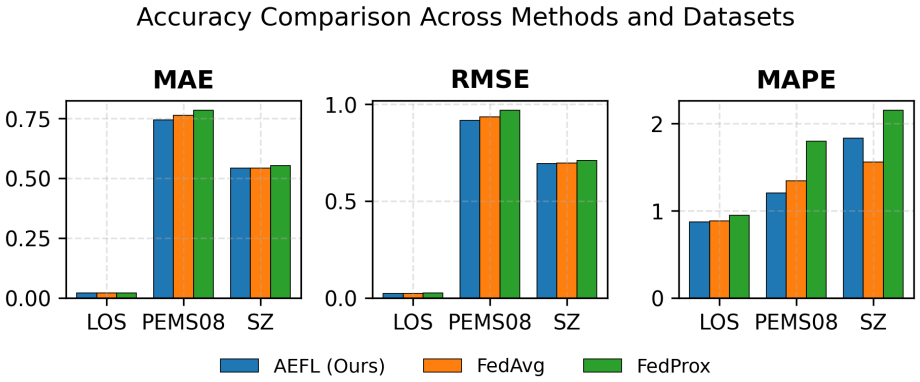

Figure 4: Accuracy comparison (MAE, RMSE, MAPE) across datasets and methods.

6.2.1 Interpretation of Accuracy Outcomes

Across all datasets, AEFL performs competitively and, in some cases, outperforms both baselines. In PeMSD8, AEFL achieves the lowest error across all metrics. Moderate heterogeneity produces clients whose contributions vary in quality, and AEFL preferentially

selects stable contributors. This filtering smooths local fluctuations and yields a more stable optimisation path across rounds. The effect is most visible in MAE, where the improvement over FedAvg is consistently maintained.

In SZ-Taxi, AEFL achieves accuracy close to FedAvg, with small improvements in MAE and RMSE. The noisier dynamics limit the amount of stable information available each round, so gains remain modest. However, maintaining parity without destabilising performance indicates that adaptivity does not sacrifice accuracy even under volatile traffic patterns.

In Los-Loop, all modes converge to nearly identical predictions due to smooth and regular traffic behaviour. The lightweight GRU model captures dominant temporal dynamics, and limited client diversity means AEFL cannot meaningfully reshape the aggregation trajectory. These results confirm that AEFL does not degrade performance in low-variability regimes.

Overall, these results support **Objective 2**: adaptive scheduling can improve convergence in heterogeneous environments without harming accuracy when workloads are uniform.

6.3 Differential Privacy Results

Differential privacy was evaluated using Gaussian noise from $\sigma = 0$ to 0.2. Figures 5 and 6 show accuracy and energy behaviour.

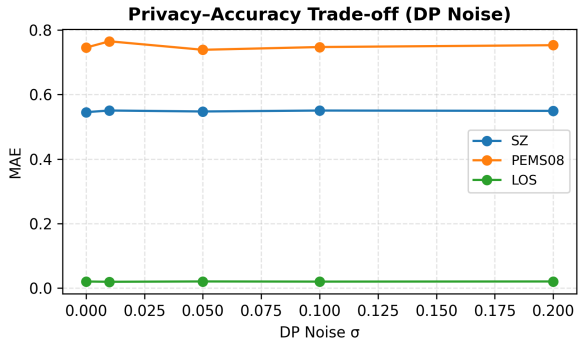

Figure 5: Privacy–accuracy trade-off (MAE vs DP noise σ).

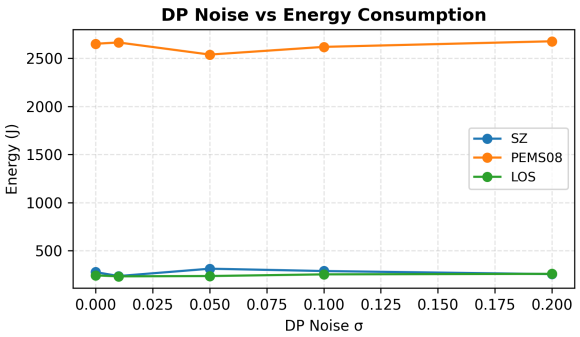

Figure 6: Energy consumption under DP noise across datasets.

To show how differential privacy affects AEFL relative to its non-private setting, Table 8 consolidates accuracy and energy values for AEFL with and without noise across all datasets.

6.3.1 Interpretation of DP Accuracy Behaviour

Accuracy remains stable for noise levels up to $\sigma = 0.1$. Beyond this threshold, errors increase gradually but remain within acceptable bounds. The GRU model has relatively few parameters, reducing sensitivity to injected noise. This suggests that lightweight architectures can sustain moderate privacy protection without substantial loss in predictive performance. Minor dataset-level fluctuations are consistent with typical DP behaviour, where noise impact varies with local data scale and variance.

Table 8: Impact of Differential Privacy on AEFL Accuracy and Energy

Dataset	DP Level	MAE	RMSE	MAPE	Energy (J)
SZ	Without DP	0.5419	0.6943	1.8336	272.87
	$\sigma = 0$	0.5453	0.6995	1.9081	277.43
	$\sigma = 0.01$	0.5504	0.7023	1.6029	236.32
	$\sigma = 0.05$	0.5474	0.7021	1.5087	312.65
	$\sigma = 0.1$	0.5503	0.7054	1.6482	288.45
	$\sigma = 0.2$	0.5493	0.7016	1.7637	256.40
PeMSD8	Without DP	0.7435	0.9181	1.2047	2782.63
	$\sigma = 0$	0.7453	0.9185	1.2314	2651.70
	$\sigma = 0.01$	0.7649	0.9349	1.2513	2665.13
	$\sigma = 0.05$	0.7386	0.9101	1.1917	2539.20
	$\sigma = 0.1$	0.7471	0.9223	1.2550	2619.14
	$\sigma = 0.2$	0.7528	0.9281	1.2333	2677.49
Los	Without DP	0.01946	0.02279	0.8692	236.26
	$\sigma = 0$	0.02053	0.02339	0.9171	243.35
	$\sigma = 0.01$	0.01968	0.02297	0.8799	233.27
	$\sigma = 0.05$	0.02072	0.02445	0.9260	236.41
	$\sigma = 0.1$	0.02021	0.02365	0.9039	253.77
	$\sigma = 0.2$	0.02065	0.02342	0.9232	259.22

6.3.2 Energy Behaviour Under Differential Privacy

Energy consumption remains close to the non-private baseline across all noise levels. The privacy mechanism is applied once after local optimisation and introduces minimal computational overhead. Differences in energy are small relative to the overall scale of each dataset. Small increases or decreases across σ values likely reflect round-to-round variance rather than systematic DP effects. These findings support **Objective 4**: differential privacy can be integrated with limited resource cost while maintaining stable behaviour across heterogeneous clients.

6.4 Combined Interpretation of Accuracy, Energy and Privacy

Overall, AEFL balances resource efficiency, predictive performance and privacy in a consistent manner. In SZ-Taxi, strong heterogeneity leads to energy savings with stable accuracy. In PeMSD8, accuracy improvements occur even when energy differences are modest, suggesting that client selection stabilises training in mixed-quality settings. In Los-Loop, workloads are uniform and AEFL behaves similarly to the baselines, indicating that adaptivity does not add overhead in less demanding scenarios. The privacy results show that moderate noise levels impose limited cost in both accuracy and energy, meaning that privacy can be added without undermining the benefits of adaptive scheduling.

6.5 Discussion

AEFL adapts effectively to dataset characteristics. Benefits are most apparent when client workloads vary significantly, as adaptive scheduling reduces unnecessary updates and improves aggregation stability. In uniform environments such as Los-Loop, AEFL converges to baseline behaviour, confirming that it does not introduce unnecessary overhead.

This adaptive neutrality is important for real deployments because it ensures that AEFL does not worsen performance when device conditions are favourable.

Differential privacy experiments further show that privacy protection can be added with minimal impact on accuracy and energy, which is important in smart-city environments with regulatory constraints. The stability of the privacy curves suggests that lightweight models remain resilient to injected noise even in highly dynamic datasets. The simplified energy model does not capture hardware-level variation, but it provides consistent comparative trends suitable for early-stage evaluation. Although compression modules are implemented, they are not evaluated extensively in this study, since the primary focus is adaptive scheduling and privacy. Overall, AEFL demonstrates robustness across diverse traffic environments and provides a practical approach to sustainable and privacy-aware federated learning.

7 Conclusion and Future Work

7.1 Conclusion

This study examined how federated learning can be adapted for heterogeneous and resource-constrained smart-city environments by developing the Adaptive Energy-Aware Federated Learning (AEFL) framework. The system combines lightweight GRU-based traffic forecasting, client-side energy estimation, adaptive participation and differential privacy, supported by an S3-based coordination layer that reflects common cloud-edge deployments. Compression mechanisms are included as optional components, but are not evaluated in depth and are therefore treated as supporting rather than central features.

Experiments on the Los-Loop, PeMSD8 and SZ-Taxi datasets show that AEFL achieves accuracy comparable to FedAvg and FedProx while reducing energy consumption in settings with noticeable workload variation, particularly in SZ-Taxi. In more uniform environments such as Los-Loop, AEFL converges to behaviour similar to the baselines, indicating that adaptive scheduling does not introduce unnecessary instability. Differential privacy experiments further show that moderate noise levels preserve accuracy and introduce only small energy changes, suggesting that privacy can be integrated without compromising overall performance. These findings support the broader aim of developing a more sustainable and practical approach to federated traffic prediction.

The evaluation is limited by reliance on containerised clients and software-based energy estimation, which approximate but cannot fully capture the irregularities of physical devices, network contention or battery-related constraints. Nevertheless, the results indicate that adaptive scheduling can provide meaningful benefits in realistic scenarios where device conditions are uneven and data availability fluctuates.

7.2 Future Work

Future work can extend AEFL by evaluating it on real embedded or battery-powered hardware. This would provide more accurate energy measurements and reveal behaviours that container-based simulations cannot capture. Another direction is to improve adaptivity through predictive modelling of client behaviour, allowing the system to anticipate changes in bandwidth or energy availability rather than only reacting to them.

The compression module also offers several opportunities for further study. Dynamically adjusting sparsity levels or quantisation scales during training may reduce com-

munication cost more effectively, particularly in bandwidth-limited settings. Privacy mechanisms could be refined using adaptive noise scaling or personalised privacy budgets to better balance protection and predictive performance. Finally, future work could integrate more expressive spatio-temporal models within the same adaptive framework to evaluate whether selective participation can offset the additional computation of deeper architectures and to further study the trade-offs that shape sustainable federated learning in intelligent transportation systems.

References

Al-Huthaifi, R., Li, T., Al-Huda, Z., Huang, W., Luo, Z. and Xie, P. (2024). Fedgode: Secure traffic flow prediction based on federated learning and graph ordinary differential equation networks, *Knowledge-Based Systems* **299**(1): 1–14.

Chen, X., Chen, Y., Saunier, N. and Sun, L. (2021). Scalable low-rank tensor learning for spatiotemporal traffic data imputation, *Transportation Research Part C: Emerging Technologies* **129**: 103226.

Deng, J. and Shen, G. (2022). Federated learning-based privacy-preserving traffic flow prediction scheme for vanets, *2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE)*, Shenzhen, China, pp. 374–378.

Domini, D., Erhan, L., Aguzzi, G., Gavallaro, L., Zenoozi, A. D., Liotta, A. and Viroli, M. (2025). Sparse self-federated learning for energy efficient cooperative intelligence in society 5.0, `https://arxiv.org/pdf/2507.07613.pdf`.

Feng, X., Sun, H., Liu, S., Guo, J. and Zheng, H. (2024). Federated meta-learning on graph for traffic flow prediction, *IEEE Transactions on Vehicular Technology* **73**(12): 19526–19538.

Feng, Y. and Qian, Q. (2025). Ppstsl: A privacy-preserving dynamic spatio-temporal graph data federated split learning for traffic forecasting, *Information Fusion* **121**(1): 1–19.

Jagadish, H. V., Gehrke, J., A., L., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R. and Shahabi, C. (2014). Big data and its technical challenges, *Communications of the ACM* **57**(7): 86–94.

Johnson, B. and Geller, M. (2025). Meta-federated learning: A novel approach for real-time traffic flow management, `https://arxiv.org/pdf/2501.16758.pdf`.

Khatua, S., De, D., Maji, S., Maity, S. and Nielsen, I. E. (2024). A federated learning model for integrating sustainable routing with the internet of vehicular things using genetic algorithm, *Decision Analytics Journal* **11**(1): 1–16.

Li, T., Sahu, A. K., Z, M., Sanjabi, M., Talwalkar, A. and Smith, V. (2020). Federated optimization in heterogeneous networks, `https://arxiv.org/pdf/1812.06127`.

Li, Y., Yu, R., Shahabi, C. and Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, `https://arxiv.org/pdf/1707.01926.pdf`.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S. and Arcas, B. A. (2023). Communication-efficient learning of deep networks from decentralized data, `https://arxiv.org/pdf/1602.05629`.

Qi, T., Chen, L., Li, G., Li, Y. and Wang, C. (2023). Fedagcn: A traffic flow prediction framework based on federate learning and asynchronous graph convolutional network, *Applied Soft Computing* **138**(1): 1–11.

Sepasgozar, S. S. and Pierre, S. (2022). Fed-ntp: A federated learning algorithm for network traffic prediction in vanet, *IEEE Access* **10**(1): 119607–119616.

Wu, K., Hao, F., Yao, R., Li, J., Min, G. and Kuznetsov, S. O. (2025). Traffic prediction based on formal concept-enhanced federated graph learning, *IEEE Transactions on Intelligent Transportation Systems* **26**(5): 6936–6948.

Yuan, X., Chen, J., Yang, J., Zhang, N., Yang, T., Han, T. and Taherkordi, A. (2023). Fedstn: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction, *IEEE Transactions on Intelligent Transportation Systems* **24**(1): 8738–8748.

Zhang, M., Li, X., Xu, B., Chen, M. and Gong, Y. (2025). Channel-independent federated traffic prediction, `https://arxiv.org/pdf/2508.04517.pdf`.

Zheng, Y., Liu, Y., Yuan, J. and Xie, X. (2011). Urban computing with taxicabs, *UbiComp '11: Proceedings of the 13th international conference on Ubiquitous computing*, ACM, Beijing, China, pp. 89–98.