Advances in Artificial Intelligence and Machine Learning; Research 5 (4) 4710-4731 Received 15-09-2025; Accepted 10-12-2025; Published online 16-12-2025

Meta-Bayesian Reinforcement Learning for Adaptive and Energy-Efficient Client Scheduling in Federated Traffic Monitoring Systems

Mohammed Ibrahim Altwijri maltwijri@kau.edu.sa
Department of Computer Science,
Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

Ahmed A. A. Gad-Elrab aaahmad4@kau.edu.sa
Department of Computer Science,
Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

Corresponding Author: Ahmed A. A. Gad-Elrab.

Abstract

Federated learning makes it possible for different devices to work together to train a model without needing to share the original data, which is why it is useful for smart traffic systems. However, static client-selection policies fail under the dynamic, non-IID, and energy-limited conditions typical of urban traffic environments. This paper proposes a federated meta-Bayesian reinforcement learning framework that integrates Bayesian meta-learning with reinforcement scheduling to adaptively select clients in traffic intelligence applications. The system updates hierarchical priors from distributed traffic data streams and optimizes client participation policies using a multi-objective reward function balancing prediction accuracy, energy efficiency, and latency constraints. In our traffic-adapted experimental setup using non-IID Fashion-MNIST (F-MNIST) data (simulating heterogeneous traffic patterns with two shards per client), The proposed framework achieves a mean accuracy of 0.84 with an 80% reduction in resource costs compared to conventional federated learning. The method provides an adaptive, energy-efficient, and self-optimizing client scheduling mechanism specifically designed for real-time federated traffic monitoring systems.

Keywords: Federated learning, Meta-Bayesian optimization, Reinforcement learning, Client scheduling, Traffic intelligence, Energy efficiency.

1. INTRODUCTION

The Internet of Things (IoT) enables continuous, real-time connectivity between heterogeneous devices [1–3]. At the same time, unmanned aerial vehicles (UAVs) and other aerial platforms receive

Citation: Mohammed Ibrahim Altwijri and Ahmed A. A. Gad-Elrab. Meta-Bayesian Reinforcement Learning for Adaptive Client Scheduling. **Advances in Artificial Intelligence and Machine Learning**. 2025;5(4):261. https://dx.doi.org/10.54364/AAIML.2025.54261

specialized communication and control support from low-altitude intelligent networks (LAIN) [4, 5], On the other hand, Vehicle-to-Everything (V2X) technologies establish cooperative connectivity between infrastructure and vehicles across the road ecosystem [6–9]. The substantial convergence of these three pillars is speeding up the shift of urban transportation toward an air–ground, integrated, intelligent monitoring paradigm [10, 11]. According to this paradigm, the basis of smart-city infrastructure for the monitoring and analysis of traffic movement is a cooperative sensing fabric composed of unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs), which offers multimodal data collection and three-dimensional coverage [12–14].

On each section of the road, UGVs and UAVs equipped with deep learning models that have already been trained are often deployed [15, 16]. While UGVs equipped with onboard sensors track fine-grained, microscopic indicators—like queue lengths and vehicle speeds—in real time, UAVs capture macroscopic conditions from above, such as incident detection and congestion patterns over an area. These systems use coordinated air-ground operations to offer a vast, multi-layered monitoring network [17–19].

Intelligent traffic monitoring aims to improve network efficiency, reduce crash risk, and provide quick, accurate information for emergency response and operational control [20]. To accomplish this goal, multidimensional sensing and low-latency situational inference must be combined. In fact, UGVs give high-resolution, close-range observations at critical nodes; UAVs provide rapid, wide-area reconnaissance of global traffic patterns; and edge computing combined with federated learning enables collaborative processing at the network edge. Together, these components enable data-driven decisions for personalized route guidance, accelerated incident response, and ideal signal timing [21].

The bandwidth that results from clients sending raw sensing data to a centralized server for training can overload wireless airwaves and pose serious privacy issues [22, 23]. Sensitive information, such as regional flow trends and accident distributions, may unintentionally be disclosed by operational road data. Additionally, prolonged high-volume transmission significantly raises mobile platforms' (like UAVs') energy demand, hastening battery depletion [24]. Device endurance deteriorates under high workloads, and in severe situations, monitoring may stop, endangering long-term stability and uninterrupted functioning.

In light of this, adaptive client scheduling under non-IID, time-varying situations, and stringent energy/latency budgets is the main problem. Conventional selection rules (or static Bayesian predictors) are myopic and deteriorate as conditions change. Clients vary in terms of data distributions, mobility, channels, and residual batteries, and these parameters drift over hours and regions. We address this by proposing a Meta-Bayesian Reinforcement Learning (MBRLFLI) scheduler that (i) learns an energy-aware, long-horizon participation policy with reward and (ii) uses hierarchical Bayesian meta-learning with task-conditioned priors updated online to deliver calibrated, uncertainty-aware estimates of each client's marginal contribution. The main innovation of the suggested method for IoT traffic intelligence is the self-learning, air-ground federated scheduler that is produced by combining predictive Bayesian optimization with policy reinforcement. This scheduler explicitly respects energy and delay restrictions while adapting to regime changes.

Federated learning (FL) addresses these issues by implementing model updates on-device and sending just parameter deltas rather than raw data, providing a bandwidth-efficient and privacy-preserving substitute for centralized training [25, 26]. Sensitive information never leaves the source in this paradigm, reducing the danger of leakage and significantly reducing backhaul usage—a benefit that becomes crucial in the presence of enormous, regularly updated traffic data streams [27, 28]. FL enhances end-to-end communication efficiency in dense urban deployments by reducing uplink payloads and localizing computing, which conforms to the limitations of air-ground sensing platforms.

However, significant variation in sensing modalities, energy constraints, and link circumstances must be addressed for real-world FL over UAVs and UGVs. Which clients train when is crucial to overall performance because unrestricted involvement can weaken learning benefits, waste energy, and undermine convergence. Device-side costs (energy, storage, and opportunity costs) also influence participation, thus clients do not voluntarily enroll in training and frequently need to be motivated [29–31]. The model owner cannot directly monitor the private expenses or data quality of each client, therefore creating a fair and efficient incentive system is a challenging task[32].

To address these gaps, we propose an air–ground cooperative Meta-Bayesian Reinforcement Learning (MBRLFLI) framework that unifies hierarchical Bayesian meta-learning with an energy-aware client-scheduling policy. On the prediction side, we replace static Bayesian scoring with a hierarchical, task-conditioned Gaussian-process (GP) meta-learner that is updated online to track regime shifts (e.g., time-of-day or region changes), yielding calibrated, uncertainty-aware estimates of each client's marginal contribution. On the decision side, a reinforcement learning (RL) agent selects the participation set by maximizing a long-horizon reward that explicitly trades off model improvement, energy, and delay. This unified design is well matched to urban traffic monitoring, where UAVs offer wide-area aerial context and UGVs provide high-resolution ground details: the meta-Bayesian layer anticipates which devices are most complementary under non-IID, time-varying conditions, while the RL policy schedules them to meet energy/latency budgets. The learnt contribution signals can be used with an optional, differentiated incentive layer to provide priority to high-quality data sources, guaranteeing incentive effectiveness without compromising privacy. These components work together to create a flexible, self-learning scheduler that maximizes the use of scarce energy and communication resources while minimizing non-IID effects and preserving high model performance.

By adapting to non-IID, time-varying traffic regimes, explicitly balancing accuracy, energy, and delay, and combining with incentive mechanisms to encourage honest, cost-conscious participation, this design turns client selection from static heuristics into a self-learning, incentive-ready scheduler that enhances training quality, system energy efficiency, and the efficacy of urban traffic monitoring.

This work introduces a self-learning scheduler for federated traffic intelligence that unifies Bayesian meta-learning with reinforcement policies. Our main contributions are:

1. **Air–ground integrated FL with adaptive scheduling.** We introduce a federated framework tailored to UAV/UGV collaboration that adapts to non-IID, time-varying traffic regimes while respecting device energy and latency constraints.
2. **Hierarchical Bayesian meta-learning for contribution prediction.** We develop a task-conditioned, online-updated Bayesian predictor (e.g., GP with hyperpriors) that provides calibrated mean/uncertainty estimates of each client's marginal utility, enabling rapid re-adaptation under regime shifts.

3. **Energy-aware reinforcement client selection.** We design a long-horizon RL policy that selects client cohorts by maximizing

$$r = \lambda_C \,\Delta\text{accuracy} - \lambda_E \,\text{energy} - \lambda_D \,\text{delay},$$

explicitly trading off model improvement, energy consumption, and latency; the Bayesian posteriors serve as informative state signals.

This paper shows a single design that puts together a step-by-step guess of Bayesian input along with a reinforcement learning planner that knows about energy and delays. Even though these parts have been looked at on their own, how they work together, where guesses from the past directly help pick clients and new results change step-by-step starting points, has not been shown in past research about federated learning. This working together makes a scheduling method that changes on its own and is made for traffic watching with UAV/UGV.

2. RELATED WORK

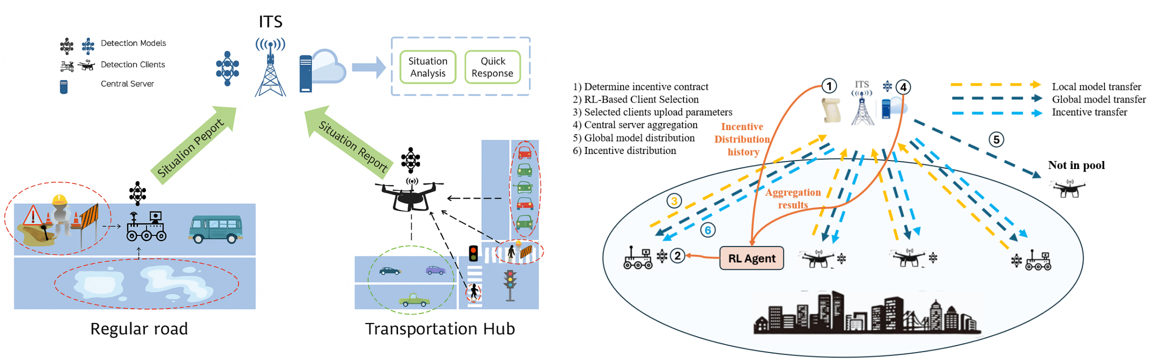
Client selection is pivotal to federated learning efficiency. Asad et al. (2021) [33], proposed a joint communication–computation resource allocation scheme that minimizes end-to-end training cost, improving resource utilization and lowering overall overhead. Qu et al. (2022) [34], introduced an online strategy based on Contextual Combinatorial Multi-Armed Bandits (CC-MAB) to select clients under budget constraints; by exploiting contextual features, their method makes more accurate choices in dynamic settings, achieving higher training performance and reduced regret.

Even with these improvements, some ways of choosing things increase gains but greatly increase the difficulty of calculations, which can make servers work harder and slower. In smart traffic watching, where main servers must deal with lots of fast-moving data and make quick choices, complex selection methods do not work well. This points out the need for a way that is both easy to use and works well, sticking to firm rules about delays and calculation limits while keeping a high standard of selection.

Older work in Federated Learning usually deals with rewards as not related to picking clients. Yet UAVs and UGVs behave as rational, self-interested agents and generally will not contribute resources without adequate compensation [35]. Contract-theoretic designs (e.g., Chen et al. (2024) [36], via Lagrange multipliers) and reputation-driven schemes (e.g., Xie et al.'s blockchain-enabled BCFR [37])—as well as hybrids combining contracts and reputation for UAV participation in FL [35, 38]—have been explored. However, these mechanisms are typically engineered *independently* of the selection policy, and the added incentive layer can introduce nontrivial computational overhead. In contrast, our proposal is *incentive-ready*: the meta-Bayesian contribution estimates feed directly into policy learning and can be used to drive differentiated rewards without bolting on a costly, separate optimization stage. The FLI framework in [39] introduced a Bayesian incentive model that improved energy efficiency but lacked adaptability. This study advances this design by embedding meta-Bayesian reasoning and reinforcement learning (RL) into the scheduling process, producing a self-adaptive system that learns client reliability and resource cost online.

However, in the existing studies, client selection and incentive design are largely treated as disjoint problems, often relying on static heuristics or high-complexity optimizers that struggle under non-

IID, time-varying conditions and tight latency/energy budgets. While incentive schemes are usually added after the fact, adding complexity without increasing adaptability, selection rules seldom ever introduce calibrated uncertainty about client utility into the decision loop. Furthermore, the majority of approaches optimize round-level, short-term goals rather than a long-horizon utility that simultaneously accounts for accuracy, energy, and delay. Our proposal fills these gaps by combining a *energy-aware reinforcement* scheduler that maximizes a long-term reward with *hierarchical meta-Bayesian* contribution prediction, which is updated live to account for regime transitions. $rdenotesthedifferencebetween\lambda_E$ energy$and\lambda_D$ delay. With the help of contribution posteriors, this unified, self-learning design naturally accommodates various incentives, produces lightweight, uncertainty-aware selection, and preserves real-time responsiveness in air-ground traffic monitoring.



(a) An Urban Traffic Monitoring Scenario [39]. (b) Federated learning with RL flowchart.

Figure 1: Schematic Scenario and Federated learning with RL flowcharts

2.1 Literature Comparison

To show where our new MBRLFLI setup fits with other federated learning setups, we give a general comparison to common methods like FedAvg, FedProx, q-FFL, and FLI. FedAvg and FedProx mainly work on making the system better and keeping things stable when the data is different, while q-FFL tries to be fair by changing how much each user affects the results. But these ways don't directly think about power limits, guessing how sure we are, or changing the plan as needed. FLI, on the other hand, uses a special system to encourage users that has something to do with picking users, but it doesn't use learning to improve, keep things safe, or learn from different levels of data. So, the MBRLFLI system does more than typical data mixing, using various prediction depths along with a power and time-aware strategy. Since computer power is limited and this study focuses on planning instead of data combination methods, we will compare our approach to others later.

There are some standard methods in the area of federated learning that mainly focus on maintaining similar results or equal treatment, such as FedAvg, FedProx, and q-FFL. These ways give us important starting points to understand how models change with different data but don't have ways to think about energy use or schedule clients based on how sure we are. The FLI system has a way to encourage clients based on how much they help, which has something to do with picking clients, but it doesn't have smart learning or ways to learn in a step-by-step manner. TABLE 1, gives a simple

Table 1: Qualitative comparison between representative federated learning methods.

Method	**Primary Focus**	**Handles Non-IID Data**	**Client Scheduling or Selection**	**Energy/Delay Awareness**
FedAvg	Model aggregation	Limited robustness on highly skewed data	No explicit scheduling	No
FedProx	Robust aggregation with proximal term	Improved stability under heterogeneity	No explicit scheduling	No
q-FFL	Fairness-oriented training	Balances client contributions	No explicit scheduling; focuses on fairness	No
FLI	Bayesian incentive-based contribution scoring	Handles heterogeneity through scoring	Partial scheduling via incentives	Limited
MBRLFLI (Proposed)	RL-based adaptive scheduling with Bayesian uncertainty	Explicitly models heterogeneity via hierarchical GPs	Yes – RL-driven, constraint-aware scheduling	Yes – energy and delay integrated

breakdown of the differences between these common ways and shows how our MBRLFLI system makes things better by adding step-by-step Gaussian-process modeling with a scheduler that knows about energy and delay using reinforcement learning.

Because of computer limits and this study's concentration on how clients are scheduled, not on combining rules, actually comparing all these ways will happen later. The descriptive look presented now puts MBRLFLI in the larger FL writing and makes clear its job as a scheduling-focused structure that helps, instead of swaps out, current combining math tools.

3. SYSTEM MODEL

Several phases, including real-time data gathering, data processing, and anomaly reporting, should be included in a comprehensive urban traffic monitoring scenario [40].

3.1 Scenario Overview

In the suggested configuration, each road section is registered and uniquely indexed in the city's intelligent traffic management system. Each area's flying and driving inspection tools have deep learning systems already set up that examine live incident updates and traffic using local processing power [41]. Driving tools travel set paths, constantly collecting info and images with built-in cameras and sensors. They figure out small details like how fast cars are going on average, how long people are waiting, and traffic jams in specific spots, along with spotting issues like accidents, car troubles, and when roads are blocked. To make this view from the ground better, flying tools often fly above roads, intersections, and main streets to offer full coverage and give a general sense of the situation across a large area.

When the system detects an abnormal condition, it immediately starts an event-reporting pipeline that notifies the traffic control center in a systematic manner. Each alert contains the route number, timestamp, geolocation data, and event category (e.g., congestion, crash, or closure). The intelligent transportation system (ITS) integrates these signals with historical patterns and live data to develop a response plan. When there is severe congestion, the system can rebalance flows by modifying the timing of surrounding signals (e.g., extending green phases). In order to minimize network-wide disturbance in the event of a crash, the emergency protocol sends out traffic police and rescue personnel while providing navigation services with dynamic detours. An overview of this air–ground monitoring scenario is depicted in FIGURE 1a.

During operation, clients continuously accumulate fresh imagery; because traffic conditions vary by roadway, these streams are inherently non-IID, causing a single global model to underperform uniformly across all segments [42]. To boost accuracy and adaptability, once a client's local dataset surpasses a preset threshold, the device triggers on-device training to incrementally refine its pre-trained model, improving fit to current road patterns and rare events [43]. In our proposal, these local updates are further governed by meta-Bayesian [44, 45] contribution estimates and an energy-aware RL scheduler, ensuring that adaptation occurs where it yields the greatest accuracy gain per unit energy and within latency constraints.

3.2 Modeling Foundations

This research uses common ways of thinking about how long communication takes, how much processing is needed, and how much power is used, to stay in line with what others have already found. The new part of our method is how we put these typical ideas into the tiered Bayesian meta-learner and then use them to make organized reinforcement-learning situations. This process allows for changing the schedule as things shift in a non-uniform way and as device limits change.

3.3 Federated Learning Model

Directly sharing raw monitoring data across roads risks privacy leakage and excessive wireless load. We therefore adopt a federated learning (FL) architecture in which UAVs/UGVs train *locally* and transmit only model updates. To align with the proposed meta-Bayesian reinforcement design, the framework is organized into two vertically coupled layers:

(i) Traffic Monitoring Client Layer

UAVs and UGVs on each road segment continuously acquire multimodal imagery for flow estimation and anomaly detection using a pre-trained model. As new data accumulate and resources permit, devices trigger local fine-tuning to better match current conditions. Let C_i denote a selected client with dataset $\mathcal{D}_i$ of size D_i. For classification, the local cross-entropy objective is

$$F_i(\boldsymbol{\theta}_t) \;=\; -\frac{1}{D_i}\sum_{j=1}^{D_i}\big[y_j \log(\hat{y}_j) + (1-y_j)\log(1-\hat{y}_j)\big]\,, \tag{1}$$

where $\hat{y}_j$ is the model output for sample j and y_j its label. Stochastic gradient descent (SGD) runs for L local epochs to obtain updated parameters $\boldsymbol{\theta}_{i,t}^{\mathrm{loc}}$ and the parameter delta

$$\Delta\boldsymbol{\theta}_{i,t} \;=\; \boldsymbol{\theta}_{i,t}^{\mathrm{loc}} - \boldsymbol{\theta}_t^{\mathrm{glob}}. \tag{2}$$

(ii) Central Server & MBRLFLI Scheduler Layer

At the start of round t, the server broadcasts the current global model $\boldsymbol{\theta}_t^{\text{glob}}$ and gathers lightweight client metadata (e.g., sample counts, recent loss, battery state, channel quality). A *hierarchical meta-Bayesian* predictor (e.g., GP with hyperpriors) is updated online to produce calibrated posteriors for each candidate's marginal contribution, yielding mean/uncertainty signals $\{(\mu_{i,t}, \sigma_{i,t})\}$. These signals, together with energy/link states, form the input to an *energy-aware reinforcement learning* policy π that selects a cohort S_t of size K under energy/latency budgets by maximizing the long-horizon reward

$$r_t \;=\; \lambda_C\, \Delta\text{accuracy}_t \;-\; \lambda_E\, \text{energy}_t \;-\; \lambda_D\, \text{delay}_t. \tag{3}$$

Selected clients perform local training and return $\{\Delta\boldsymbol{\theta}_{i,t}\}_{i\in S_t}$ over wireless uplinks.

Communication To support parallel, contention-free uploads from multiple roads, the system uses orthogonal frequency-division multiple access (OFDMA) [46], allocating subchannels to clients in S_t to reduce interference and latency during model-update transmission.

Aggregation and Update Upon reception, the server aggregates client deltas to form

$$\mathbf{w}_t \;=\; \lambda \sum_{i\in S_t} \Delta\boldsymbol{\theta}_{i,t}, \qquad \boldsymbol{\theta}_{t+1}^{\text{glob}} \;=\; \boldsymbol{\theta}_t^{\text{glob}} + \mathbf{w}_t, \tag{4}$$

where $\lambda > 0$ is an aggregation weight. The round concludes with (optional) differentiated incentives that can be computed directly from the meta-Bayesian contribution posteriors to encourage high-impact participation without accessing raw data.

Termination Rounds repeat until a stopping rule is met, e.g., $\left|\mathcal{F}(\boldsymbol{\theta}_{t+1}) - \mathcal{F}(\boldsymbol{\theta}_t)\right| \le \varepsilon$ for a tolerance $\varepsilon > 0$, or a preset budget/round limit is reached. Throughout training, the hierarchical priors and the RL policy co-evolve: selections generate new evidence to refine the priors, while updated posteriors improve future scheduling. This closed-loop MBRLFLI process yields adaptive, uncertainty-aware client selection that explicitly balances accuracy, energy, and delay in air–ground traffic monitoring. The federated learning process with incentives and RL scheduler is shown in FIGURE 1b.

3.4 Adapted Communication Model

We employ an OFDMA uplink with N orthogonal subchannels ($N \ge K$) of bandwidth W. To keep focus on scheduling and learning, we retain an additive Gaussian white noise (AWGN) abstraction (it has been commonly used in prior FL studies (e.g., [47, 48]) for rate calculation, while allowing the scheduler to ingest stochastic link indicators (e.g., recent SNR samples, outage flags) as *state* variables. For client C_k in round t, the achievable rate on its assigned subchannel is

$$r_k^{\text{up}}(t) \;=\; W \log_2\!\left(1 + \frac{p_k^{\text{up}}(t)\, h_k(t)}{\sigma^2}\right), \tag{5}$$

with payload $s_k^{\text{up}}(t)$, delay $\tau_k^{\text{up}}(t) = s_k^{\text{up}}(t)/r_k^{\text{up}}(t)$, and uplink energy

$$E_k^{\text{com}}(t) \;=\; p_k^{\text{up}}(t)\, \tau_k^{\text{up}}(t). \tag{6}$$

MBRLFLI integration. The meta-Bayesian layer provides calibrated contribution posteriors $(\mu_{k,t}, \sigma_{k,t})$, while the RL policy uses $(\mu_{k,t}, \sigma_{k,t})$ together with *measured* communication costs

$\big(E_k^{\mathrm{com}}(t),\ \tau_k^{\mathrm{up}}(t)\big)$ when forming the state s_t and deciding the cohort S_t. Safety filters enforce per-round latency and energy caps during action selection.

3.5 Adapted Computation Model

Local training on C_k for L epochs over $D_k(t)$ samples at CPU frequency $f_k(t)$ incurs

$$\tau_k^{\mathrm{cal}}(t) \;=\; \frac{C_k^{\mathrm{yc}}\, D_k(t)}{f_k(t)} \tag{7}$$

$$E_k^{\mathrm{cal}}(t) \;=\; \kappa\, C_k^{\mathrm{yc}}\, D_k(t)\, f_k^2(t) \tag{8}$$

where C_k^{yc} is cycles-per-sample and κ the effective switching capacitance. Devices expose lightweight compute telemetry (e.g., $f_k(t)$ limits, battery state), which the policy maps to expected compute costs $\big(L\,\tau_k^{\mathrm{cal}}(t),\ L\,E_k^{\mathrm{cal}}(t)\big)$. The meta-Bayesian predictor conditions on such features (and task context like time-of-day/region) to refine contribution estimates under drift, enabling *cost-informed* yet *uncertainty-aware* selection.

3.6 Optimization Problem Formulation

Given round t cohort S_t (with $|S_t| = K$), the per-client totals are

$$\tau_k(t) = L\,\tau_k^{\mathrm{cal}}(t) + \tau_k^{\mathrm{up}}(t), \tag{9}$$

$$E_k(t) = L\,E_k^{\mathrm{cal}}(t) + E_k^{\mathrm{com}}(t). \tag{10}$$

Let $Y_k(t)$ denote the (latent) marginal improvement from including C_k in round t. The meta-Bayesian layer produces an online-updated posterior $p(Y_k(t) \mid \text{features})$ summarized by mean/uncertainty $(\mu_{k,t}, \sigma_{k,t})$. We maximize long-horizon performance with an energy-/delay-aware reward:

$$r_t \;=\; \lambda_C \Delta \mathrm{Acc}_t \;-\; \lambda_E \sum_{k\in S_t} E_k(t) \;-\; \lambda_D \max_{k\in S_t} \tau_k(t), \tag{11}$$

subject to per-round safety constraints

$$\begin{aligned} &E_k(t) \le E_{\max},\;\; \tau_k(t) \le \tau_{\max},\;\; \forall k \in S_t,\\ &|S_t| = K. \end{aligned} \tag{12}$$

RL policy. The scheduler learns a policy $\pi(S_t \mid s_t)$ over combinatorial actions using the state

$$\begin{aligned} s_t = \Big\{ &(\mu_{k,t}, \sigma_{k,t}),\ E_k^{\mathrm{com}}(t),\ \tau_k^{\mathrm{up}}(t),\ E_k^{\mathrm{cal}}(t),\ \tau_k^{\mathrm{cal}}(t),\\ &\text{battery/channel/meta}\Big\}_k . \end{aligned} \tag{13}$$

A projection layer enforces (12) during selection. This yields a *closed-loop* system: selections provide new observations to update hierarchical priors; refined posteriors improve subsequent decisions, ensuring adaptation to non-IID drift while explicitly balancing accuracy, energy, and latency.

To obtain a round-wise solvable program, we introduce binary decision variables

$$x_{k,t} \in \{0,1\}, \quad x_{k,t} = 1 \iff \text{client } C_k \text{ is selected in round } t,$$

and treat $\{Y_k(t), E_k(t), \tau_k(t)\}$ as known per-round signals (e.g., predicted contribution from the meta-Bayesian layer and measured/specified costs). The linear (mixed-integer) program reads:

$$\text{(P1-LP)} \quad \max_{\{x_{k,t}\}} \quad \sum_{t=1}^{T}\sum_{k=1}^{N} Y_k(t)\, x_{k,t} \tag{14}$$

$$\text{s.t.} \quad \sum_{k=1}^{N} x_{k,t} = K, \qquad \forall\, t = 1, \ldots, T, \tag{15}$$

$$E_k(t) \le E_{\max} + M_E\,(1 - x_{k,t}), \quad \forall\, k, t, \tag{16}$$

$$\tau_k(t) \le \tau_{\max} + M_\tau\,(1 - x_{k,t}), \quad \forall\, k, t, \tag{17}$$

$$x_{k,t} \in \{0,1\}, \qquad \forall\, k, t. \tag{18}$$

Here, (14) maximizes the cumulative contribution, (15) enforces a cohort of size K each round, and (16)–(17) activate energy and latency limits only when a client is selected ($x_{k,t} = 1$). The constants M_E and M_τ are sufficiently large "big-M" values that render the constraints inactive when $x_{k,t} = 0$.

Remarks (MBRLFLI compatibility). In our proposal, $Y_k(t)$ can be set to the meta-Bayesian posterior mean (optionally risk-adjusted), while $E_k(t)$ and $\tau_k(t)$ come from the communication/computation models. Solving (P1-LP) yields a deterministic top-K cohort under per-client budgets; the RL policy can be viewed as learning a fast, online approximation to this selection under uncertainty and drift.

Compared with static, round-myopic selection, this MBRLFLI formulation: (i) *learns to predict* contribution under regime shifts via hierarchical priors, (ii) *optimizes long-term utility* with explicit energy/latency penalties, and (iii) preserves practicality by using measurable comm/compute signals and safety filters for real-time operation on UAV/UGV fleets.

4. PROPOSED FRAMEWORK: META-BAYESIAN RL FOR ADAPTIVE CLIENT SCHEDULING

This section presents the complete framework of our *Meta-Bayesian Reinforcement Learning* (MBRLFLI) scheduler for federated traffic intelligence. The design unifies (i) a **hierarchical Bayesian** contribution predictor that adapts online to non-IID regime shifts with (ii) an **energy-aware RL** policy that selects client cohorts under explicit energy/latency constraints.

4.1 Overview and Design Goals

Air–ground traffic monitoring exhibits non-stationary data (region/time-of-day/incident regimes), heterogeneous devices (UAV/UGV), and strict energy/latency budgets. Our goals are:

1. **Adaptivity**: track distribution drift via hierarchical priors and task conditioning.
2. **Resource-awareness**: schedule clients with explicit energy/delay penalties.

3. **Uncertainty usage**: propagate predictive uncertainty into decisions.
4. **Real-time feasibility**: ensure light per-round overhead with safety filters.

Because we have limited room, the paper does not have a picture showing the process flow. Instead, we give a written description, step by step, of the whole process, which includes getting data, using Bayesian updating, scheduling with RL, predicting safety, and combining things, to explain all the methods we used.

4.2 Two-Layer Architecture

Client layer (UAV/UGV). Devices perform on-device inference/learning, expose lightweight telemetry (battery state, CPU cap, SNR), and, when selected, run L local epochs to produce model deltas $\Delta\boldsymbol{\theta}_{i,t}$.

Server layer (MBRLFLI scheduler). Each round t: (i) update hierarchical priors and compute contribution posteriors $(\mu_{i,t}, \sigma_{i,t})$; (ii) form the RL state from Bayesian signals and comm/compute costs; (iii) select the cohort S_t; (iv) aggregate updates and refresh the global model.

4.3 Meta-Bayesian Contribution Predictor

We model the *marginal contribution* $Y_i(t)$ (e.g., expected loss reduction if client i participates at round t) as a task-conditioned stochastic process. Let $x_{i,t}$ denote client/task features (region, time window, data size, last-round loss, battery, SNR, etc.). A hierarchical Gaussian-process (HGP) prior

$$Y_i(t) \mid \tau \sim \mathcal{GP}\big(m_\tau(x),\ k_\tau(x,x')\big), \tag{19}$$

$$(m_\tau, k_\tau) \mid \phi \sim \text{HyperPrior}(\phi), \tag{20}$$

shares information across tasks τ (e.g., region/time) via hyperparameters ϕ. Online variational updates yield per-candidate posteriors

$$p\big(Y_i(t) \mid \mathcal{D}_{1:t}\big) \ \rightsquigarrow \ (\mu_{i,t},\ \sigma_{i,t}), \tag{21}$$

where $\mu_{i,t}$ summarizes predicted utility and $\sigma_{i,t}$ captures epistemic uncertainty. These summaries enter the policy as risk-aware signals (e.g., *upper/lower confidence* scores $\mu_{i,t} \pm \beta\sigma_{i,t}$).

The way of organizing Gaussian-process models in levels is chosen because it easily catches changes that depend on the situation, like those that come from different areas, times in the day, or ways of sensing. Even though thoroughly checking if it always works or how much it might fail is not something we will do here, these leveled GPs have well-known math qualities—like smooth results, managing differences, and sharing info across tasks—that show they are good for changing client-scheduling jobs.

4.4 RL Policy and Constraint Handling

The P1-LP setup shows the best possible goal for picking customers in a set way when considering power use and delays. The learning-based planner works as a helpful way to estimate this goal while running. Its safety feature makes sure the same limits from P1-LP are followed, keeping the ideas consistent between the plan and how decisions are actually made.

4.5 Energy-Aware RL Scheduling

We define the per-round state s_t by concatenating Bayesian signals and measurable costs:

$$s_t = \Big\{(\mu_{i,t}, \sigma_{i,t}),\ E_i^{\mathrm{com}}(t),\ \tau_i^{\mathrm{up}}(t),\ E_i^{\mathrm{cal}}(t),\ \tau_i^{\mathrm{cal}}(t),\ \text{battery}_i(t),\ \mathrm{SNR}_i(t),\ \text{region/time context}\Big\}_{i=1}^{N}. \tag{22}$$

where, a_t is a subset S_t of size K (combinatorial).

So, the reward trades-off bettween accuracy, energy, and delay is defined as follows:

$$r_t \;=\; \lambda_C \Delta \mathrm{Acc}_t \;-\; \lambda_E \sum_{i\in S_t} E_i(t) \;-\; \lambda_D \max_{i\in S_t} \tau_i(t). \tag{23}$$

Also, the safety system shows possible actions within the allowed group:

$$E_i(t) \le E_{\max}, \quad \tau_i(t) \le \tau_{\max}, \quad \forall i \in S_t, \qquad |S_t| = K. \tag{24}$$

4.6 Learning Loop and Closed-Form Interfaces

Communication and computation costs follow the standard models:

$$r_i^{\mathrm{up}}(t) = W \log_2\!\left(1 + \frac{p_i^{\mathrm{up}}(t) h_i(t)}{\sigma^2}\right), \quad \tau_i^{\mathrm{up}}(t) = \frac{s_i^{\mathrm{up}}(t)}{r_i^{\mathrm{up}}(t)}, \tag{25}$$

$$E_i^{\mathrm{com}}(t) = p_i^{\mathrm{up}}(t)\tau_i^{\mathrm{up}}(t), \tag{26}$$

$$\tau_i^{\mathrm{cal}}(t) = \frac{C_i^{\mathrm{yc}} D_i(t)}{f_i(t)}, \quad E_i^{\mathrm{cal}}(t) = \kappa C_i^{\mathrm{yc}} D_i(t) f_i^2(t), \tag{27}$$

$$\tau_i(t) = L\tau_i^{\mathrm{cal}}(t) + \tau_i^{\mathrm{up}}(t), \quad E_i(t) = LE_i^{\mathrm{cal}}(t) + E_i^{\mathrm{com}}(t). \tag{28}$$

These values are used for both the Bayesian qualities and the RL situation, making sure choices are made with *cost-informed*. Algorithm 1 shows the MBRLFLI Scheduling per FL Round. In this Algorithm, the HGP_Update function uses gradual changes to statistical guesses to refresh layered Gaussian-process predictions using new info about the users. The $Proj\_F$ tool makes sure things are doable by putting any possible action onto the list of user choices that fit within energy rules, time limits, and unchanging group size. These parts make sure planning stays steady and follows rules.

Algorithm 1 MBRLFLI Scheduling per FL Round

Inputs: global model $\boldsymbol{\theta}_t^{\text{glob}}$, candidate set C_t, priors (ϕ_t), budgets $(E_{\max}, \tau_{\max})$
Observe features $x_{i,t}$ and costs $\left(E_i^{\text{com}}(t), \tau_i^{\text{up}}(t), E_i^{\text{cal}}(t), \tau_i^{\text{cal}}(t)\right)$ for $i \in C_t$
Meta-Bayes update: posterior $(\mu_{i,t}, \sigma_{i,t}) \leftarrow \text{HGP\_Update}(\phi_t, \mathcal{D}_{1:t}, x_{i,t})$
Form state s_t from $\{(\mu_{i,t}, \sigma_{i,t})\}$ and cost telemetry
Sample provisional cohort $\tilde{S}_t \sim \pi(\cdot \mid s_t)$ (actor)
Safety projection: $S_t \leftarrow \text{Proj}_{\mathcal{F}}(\tilde{S}_t; E_{\max}, \tau_{\max}, K)$
Broadcast $\boldsymbol{\theta}_t^{\text{glob}}$ to S_t; clients run L local epochs and return $\{\Delta\boldsymbol{\theta}_{i,t}\}_{i \in S_t}$
Aggregate: $\mathbf{w}_t = \lambda \sum_{i \in S_t} \Delta\boldsymbol{\theta}_{i,t}$; update $\boldsymbol{\theta}_{t+1}^{\text{glob}} = \boldsymbol{\theta}_t^{\text{glob}} + \mathbf{w}_t$
Compute reward r_t via (23); update critic/actor using (s_t, S_t, r_t, s_{t+1})
Append new outcomes to $\mathcal{D}_{1:t}$; update hyperparameters ϕ_{t+1}

4.7 Complexity and Deployment

Per round, MBRLFLI adds (i) incremental HGP updates on compact features and (ii) a forward pass of the actor/critic with a safety projection. These are lightweight relative to model training/communication and scale linearly with the number of candidates. The framework supports asynchronous participation and naturally integrates with OFDMA scheduling.

4.8 Incentive-Ready Interface

Because the meta-Bayesian layer outputs calibrated contribution posteriors, differentiated incentives can be computed directly from $(\mu_{i,t}, \sigma_{i,t})$ (e.g., risk-adjusted payments) without accessing raw data or solving a separate heavy optimization. This preserves privacy while encouraging high-impact participation.

5. EXPERIMENTS AND RESULT ANALYSIS

This section evaluates the proposed Meta-Bayesian Reinforcement Learning (MBRLFLI) client scheduling framework using the Fashion-MNIST (F-MNIST) dataset [49]. The F-MNIST dataset offers diverse image categories such as apparel and footwear, making it appropriate for simulating non-IID data distributions encountered in real-world federated traffic intelligence scenarios. Different parts of roads might have different problems, traffic jams, or traffic behaviors. We can split up customers using labels to copy this variety well.

Our method uses 80% of the data to train models locally on each customer's device and 20% to check the main model. We test our new MB-RL method against regular federated learning methods to see how much better it is in accuracy, scheduling, and cost. The data shows MB-RL makes the model work better when things change, power is limited, and data is different by learning how to pick customers wisely. The meta-Bayesian layer makes scheduling much more stable by using past data about how well things worked.

5.1 Experimental Rationale

The goal of the tests is to see what the scheduling system does with different types of changing data, not to recreate images of real traffic. This explains why Fashion-MNIST is only used to create varied data groups, which is a standard practice in studies about federated optimization and scheduling. The scheduler uses details like predicted uncertainty, loss values, energy consumption, and communication delays, rather than picture content, so using this test data is still acceptable. At a later time, actual traffic data will be implemented.

The way the test is set up uses a fairly big system to see how well the proposed scheduler works using data that changes in ways we can easily understand and with not much available to use. This is close to the ways things are normally done in studies that look at federated scheduling. More thorough testing will be done later to more clearly show that the system can handle bigger loads.

5.2 Experimental Setup

We trained a multi-layer perceptron (MLP) model with two hidden layers using the Fashion-MNIST dataset. The collection was spread out unevenly to copy differences in data between users. The method of giving each user two unique parts of the collection was utilized for dividing the data, just like in a previous study [50]. In particular, we separated the entire dataset into 200 equal-sized shards, each of which comprised data samples from a single label, after sorting it by class labels. Next, two shards were assigned at random to each client. This reduced label diversity while guaranteeing that each client had the same dataset size.

One hundred UAV clients were part of the federated learning environment. Five clients were chosen for training in each communication round. This restriction on choices helps keep communication overhead and energy use under check. TABLE 2, provides a summary of all simulation parameters, including system setting and model training. The parameter α_{t_k} stands for the adjustment amount that comes from the FLI system. In this work, it keeps the effect of contribution guesses steady on calculations involving incentives and stops sudden changes in how rewards work.

Table 2: Experimental Configuration

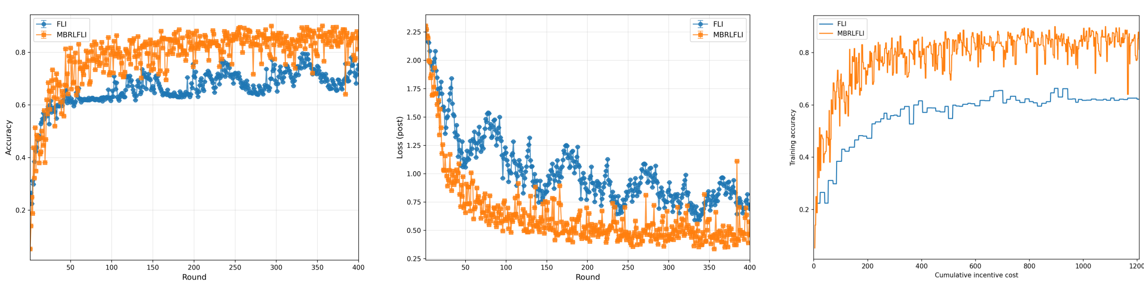
Parameter Name	**Value**
Dataset	MNIST (non-IID, 2 shards/client)
Clients	100
Model	MLP (128,256)
Rounds	400
Local Training Epochs	3
Learning Rate	0.005
Batch Size	64
Incentive Budget R_{max}	20,000
Pakcet size for calculating comm. Time and Energy	[64, 512]
Contribution Adjustment Coefficient β	0.01
$\alpha_{t_k} \in (0, 1)$	0.75
λ_C, λ_E, and λ_D	0.8, 0.1, and 0.1
Framework	PyTorch + SimPy
Baseline	FLI [39]
Metrics	Accuracy, Loss, Energy, Delay, Cost

To guarantee steady convergence throughout training, we employed a learning rate of 0.005. Every chosen client used three epochs every round to train its local model. This figure provided a trade-off between model accuracy and processing load. The amount of samples per gradient update was regulated by the batch size; larger batches increased speed but may have an impact on generalization. To limit the overall amount of rewards given to clients in each round, an incentive budget was established. The way incentive payouts were influenced by client performance was controlled by the Contribution Adjustment Factor. In order to replicate actual or hypothetical network conditions that impact model upload costs, communication time and energy characteristics were also incorporated. We conducted comparative studies with the baseline approach FLI [39], to evaluate the effectiveness of our suggested scheme.

5.3 Baseline Selection Justification

This research looks at how well the system changes customer scheduling instead of comparing different ways to combine things. The FLI way is picked as the main thing to measure against because it has similar ideas about using data and rewards, which fits with what we are trying to do. Ways like FedAvg, FedProx, and q-FFL mostly deal with how to combine information, so they do not really test how good the scheduling is. We plan to include them in more tests later on.

5.4 Results and Discussion



(a) Accuracy vs. Round (b) Post-training loss vs. round. (c) Accuracy vs. incentive cost.

Figure 2: Comparison of the accuracy, loss, and cost characteristics of the MBRLFLI and FLI models.

FIGURE 2a, sevaluates the training accuracy of the FLI and MBRLFLI algorithms on the non-IID F-MNIST dataset over 400 federated learning cycles. The baseline FLI model [39], exhibits minimal generalization under diverse clients, rising progressively until approximately round 60, then stabilizing between 0.60 and 0.70 accuracy. MBRLFLI, on the other hand, climbs more sharply in the early rounds, topping 0.80 accuracy by round 50 and continuing to converge steadily above 0.85 after round 100. Smaller changes in later stages show that the rules were better adjusted and that learning happened evenly for different types of customers. This trend shows that using a mix of general learning with reward-based planning makes things come together more quickly and dependably than just using basic general learning.

FIGURE 2b, shows the loss after training for MBRLFLI and FLI [39], across 400 times the federated process was completed. The FLI comparison group constantly loses value even after lots of training, but stays above 1.0, showing it takes longer to balance out and has more mistakes. On the other hand, MBRLFLI drops more quickly and steadily, going under 0.5 after the first 100 times and staying stable after that. This steady drop reveals how the model can change which clients are picked by using meta-Bayesian changes and reinforcement learning, leading to quicker balance and better optimization. The obvious gap between the two lines proves MBRLFLI works better than the comparison in getting to a lower stable loss and balancing out faster.

FIGURE 2c, shows how the total cost of incentives relates to how well the training works for both FLI [39], and MBRLFLI. The standard FLI gets better at training as the incentive cost goes up, but then it stops improving at 0.65, which means using incentives to make the model better does not work very well. On the other hand, MBRLFLI trains with over 0.85 accuracy while still costing much less in total. This study makes clear that the meta-Bayesian reinforcement setup gives out rewards in a better way by picking clients who help the learning improve more. The big jump in training quality without much cost shows that MBRLFLI can give a better balance between cost and training quality compared to how FLI used to work, all while keeping an eye on both performance and how well the money is spent.

(a) Average delay vs. round. (b) Average energy vs. round.

Figure 3: MBRLFLI and FLI models' performances in terms of energy and delay metrics.

FIGURE 3a, shows how long each communication step took on average for FLI and MBRLFLI over 400 steps. Both ways have times that change because of different internet connections from clients and time spent computing, but MBRLFLI usually has a little less waiting time for most steps. By using smarter ways to pick clients with better internet speed and steadier response times, MBRLFLI's ability to adjust using reinforcement learning makes the wait times less variable. Compared to FLI, which sometimes has really long wait times, MBRLFLI keeps the waiting time steadier without losing correctness. This means communication works better and different client setups are handled better in the federated setting.

FIGURE 3b, exhibits the average energy consumed during 400 training rounds for FLI and MBRLFLI. Similar fluctuations are observed between the two algorithms, which are linked to changes in client device usage and communication expenses during each cycle. MBRLFLI consistently maintains a lower average energy usage than FLI, indicating greater energy efficiency under changing consumer conditions. MBRLFLI's adaptive reinforcement policy is responsible for this advancement, as it selects clients based on both expected learning benefit and energy cost. By taking this into account, MBRLFLI offers a more favorable balance between computational efficiency and model performance, making it ideally suited to energy-conscious federated learning in real-life situations.

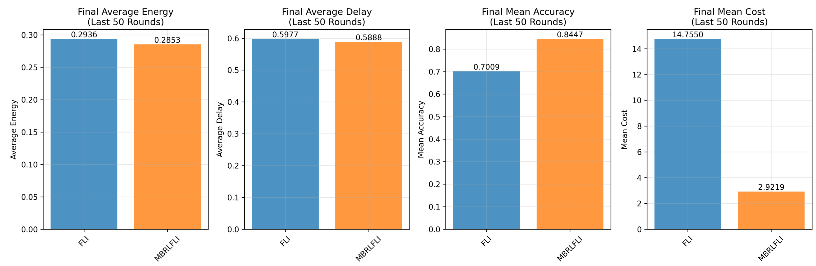


Figure 4: Final 50-round comparison for energy, delay, accuracy, and mean cost.

FIGURE 4, compares the performance of FLI and MBRLFLI across four metrics, averaged over the last 50 training rounds using the F-MNIST dataset. With a mean accuracy of 0.8447 compared to FLI which was 0.7009, MBRLFLI's classification accuracy is enhanced by 20.5%. In terms of energy efficiency, MBRLFLI uses less energy on average: 0.2853 versus 0.2936 for FLI. Furthermore, MBRLFLI has a lower average latency (0.5888) than FLI (0.5977), suggesting better responsiveness. Most significantly, under MBRLFLI, the mean cost drops by more than 80% from 14.7550 under FLI to 2.9219. These results indicate that MBRLFLI is more suitable for resource-constrained federated learning contexts because to its superior performance in terms of accuracy, energy usage, latency, and total cost.

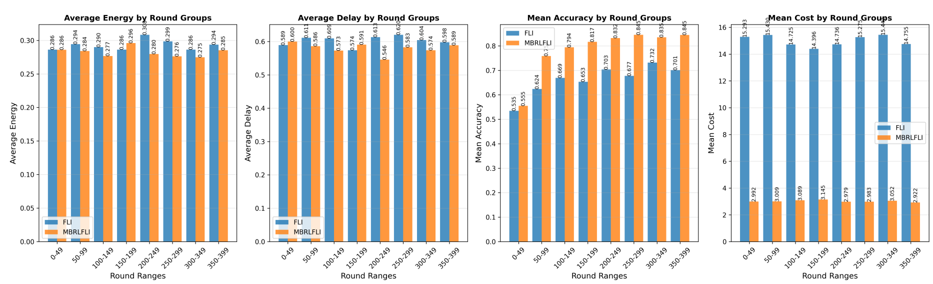


Figure 5: Mean cost per interval: MBRLFLI consistently reduces cost by ≈80%.

FIGURE 5, compares FLI and MBRLFLI performance over grouped training rounds using four metrics: energy, delay, accuracy, and cost. In every round range, MBRLFLI performs better than FLI in terms of accuracy; it starts at 0.624 in the early rounds and reaches 0.845 by the end of the range, whereas FLI only reaches a top of 0.732. Overall, MBRLFLI results in slightly lower energy and delay values, suggesting modest but consistent gains in resource efficiency. Cost reduction is significant at all points, with FLI being between 14.39 and 15.57 and MBRLFLI remaining between 2.92 and 3.14. However, the former value remains unchanged. The regular margins indicate that MBRLFLI offers long-term improvements in precision and effectiveness.

Table 3: Final 50-Round Performance Summary

Metric	FLI [39]	MBRLFLI	Improvement
Mean Accuracy	0.7009	**0.8447**	+20.5%
Mean Loss	1.12	**0.56**	-50.0%
Avg Delay	0.5977	**0.5888**	-1.5%
Avg Energy	0.2936	**0.2853**	-2.8%
Mean Cost	14.755	**2.922**	-80.2%

In TABLE 3, the proposed MBRLFLI scheme and the basic FLI approach after 50 rounds are presented. According to the data, MBRLFLI performs exceptionally well, with an average accuracy increase of 20.5% and an overall loss reduction of 501%. This results in improved model learning and a stable state of equilibrium when things are not consistent. The amount of time taken remains roughly the same, but it tends to improve by 1.5%, indicating that the RL tool does not cause any slowdown. MBRLFLI reduces power consumption, resulting in an average saving of 2.8% because it selects clients that are resourceful. MBRLFLI's biggest change is in the overall cost, but it still manages to save 80.2% by planning ahead and deciding on power limits. The proposed setup significantly improves the accuracy of things and the efficiency of resources compared to the standard method, as evidenced by the data.

6. LIMITATIONS

The limitations of this work will determine how we proceed in the future. How will that be achieved? While the process of cutting out parts and examining their effects is informative, this one requires minimal computer processing power due to the need for multiple training setups. Therefore, it is not included. The writing explains the roles of the different parts, including their special layer that uses past knowledge, their system that learns from rewards and their safe mechanism, all of which are intended to be fully tested later by cutting them off.

The system that organizes tasks uses data points such as device power usage, processing requirements, expected errors, and signal time to facilitate its use with diverse data collections. While this setup can be used for any data collection, future studies must consider testing the system with real time and location data from transportation.

Because we had limits on computer power, all tests are shown as results from just one try. Even though this way is good enough to show how the scheduling acts and what the results tend to be, later studies will use many tries to look at changes and check if the results are real.

In the end, the review has just one standard model (FLI), picked because it is like the suggested model in basic ways. In the future, tests that use more standard federated learning models will be added to make the proof stronger.

7. CONCLUSION

In this paper a meta-Bayesian reinforcement framework for adaptive client scheduling in federated traffic intelligence called MBRLFLI was presented. By combining predictive uncertainty modeling and reinforcement optimization, MBRLFLI improves the accuracy–cost ratio. The MNIST results show a significant improvement over FLI. Both cross-domain transfer and asynchronous scheduling are used. In future work, we will study the performance of using real traffic datasets. In the Future Work, the evaluation on real traffic datasets, integration of temporal traffic models, multi-UAV coordination, and asynchronous federated scheduling will be studied.

8. ACKNOWLEDGMENT

The author acknowledges the resources and assistance provided by Faculty of Computing and Information Technology, King Abdulaziz University.

References

[1] Kopetz H, Steiner W. Internet of Things. In Real-Time Systems: Design Principles for Distributed Embedded Applications. Cham: Springer International Publishing. 2022:325-341.

[2] Bello O, Zeadally S. Intelligent Device-To-Device Communication in the Internet of Things. IEEE Syst J. IEEE. 2014;10:1172-1182.

[3] Tien JM. Internet of Things Real-Time Decision Making and Artificial Intelligence. Ann Data Sci .2017;4:149-178.

[4] Hossein Motlagh NH, Taleb T, Arouk O. Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives. IEEE Internet Things J. 2016;3:899-922.

[5] Qiu J, Grace D, Ding G, Zakaria MD, Wu Q. Air-Ground Heterogeneous Networks for 5G and Beyond via Integrating High and Low Altitude Platforms. IEEE Wirel Commun. 2019;26:140-148.

[6] Wang J, Shao Y, Ge Y, R Yu. A Survey of Vehicle to Everything (v2x) Testing. Sensors. 2019;19:334.

[7] Tong W, Hussain A, Bo WX, Maharjan S. Artificial Intelligence for Vehicle-To-Everything: A Survey. In IEEE Access. 2019;7:10823–10843.

[8] Seo H, Lee KD, Yasukawa S, Peng Y, Sartori P. LTE Evolution for Vehicle-To Everything Services. IEEE Commun Mag. 2016;54:22-28.

[9] Yin Y, Liu M, Gui G, Gacanin H, Sari H. Minimizing Delay for Mimo-Noma Resource Allocation in Uav-Assisted Caching Networks. In IEEE Trans Veh Technol. 2023;72:4728-4732.

[10] Nellore K, Hancke GP. A Survey on Urban Traffic Management System Using Wireless Sensor Networks. Sensors. 2016;16:157.

[11] Liu J, Shi Y, Fadlullah ZM, Kato N. Space-Air-Ground Integrated Network: A Survey. IEEE Commun Surv Tutor. 2018;20:2714-2741.

[12] Yin Y, Gui G, Liu M, Gacanin H, Sari H, et al. Joint User Pairing and Resource Allocation in Air-To-Ground Communication-Caching-Charging Integrated Network Based on Noma. In IEEE Trans Veh Technol. IEEE. 2023;72:15819-15828.

[13] Chen Q, Meng W, Li S, Li C, Chen HH. Civil Aircrafts Augmented Space–Air–Ground integrated Vehicular Networks: Motivation Breakthrough and Challenges. IEEE Internet Things J. 2021;9:5670-5683.

[14] Cheng N, Xu W, Shi W, Zhou Y, Lu N, et al. Air-Ground Integrated Mobile Edge Networks: Architecture Challenges and Opportunities. In IEEE Commun Mag. 2018;56:26-32.

[15] Buch N, Velastin SA, Orwell J. A Review of Computer Vision Techniques for the Analysis of Urban Traffic. IEEE Trans Intell Transp Syst. 2011;12:920-939.

[16] Zhao D, Dai Y, Zhang Z. Computational Intelligence in Urban Traffic Signal Control: A Survey. IEEE Trans Syst Man Cybern C. Applications and Reviews. 2011;42:485-494.

[17] Zhang Y, Chen D, Wang S, Tian L. A Promising Trend for Field Information Collection: An Air-Ground Multi-Sensor Monitoring System. Inf Process Agric. 2018;5:224-233.

[18] Barrado C, Messeguer R, López J, Pastor E, Santamaria E, et al. Wildfire Monitoring Using a Mixed Air-Ground Mobile Network. IEEE Pervasive Comput. 2010;9:24-32.

[19] Valente J, Sanz D, Barrientos A, Cerro JD, Ribeiro Á, et al. An Air-Ground Wireless Sensor Network for Crop Monitoring. Sensors. 2011;11:6088-6108.

[20] Nigam N, Singh DP, Choudhary J. A Review of Different Components of the Intelligent Traffic Management System (ITMS). Symmetry. 2023;15:583.

[21] Latif S, Afzaal H, Zafar NA. Intelligent Traffic Monitoring and Guidance System for Smart City. In: Proceedings of the 2018 international conference on computing mathematics and engineering technologies (iCoMET). 2018:1-6.

[22] Jordan S, Moore J, Hovet S, Box J, Perry J, et al. State-of-The-Art Technologies for Uav Inspections. IET Radar Sonar Navig. 2018;12:151-164.

[23] Mentsiev AU, Guzueva ER, Magomaev TR. Security Challenges of the Industry 4.0. J Phys Conf Ser. 2020;1515:032074.

[24] Zhao H, Shi Y, Liu M, Zhu H, Xun W. Fairness Can Save Lives: A Mab Based Client Selection Strategy for Federated Learning Towards IoV Assisted ITS. IEEE Trans Veh Technol. 2024;74:5430-5441.

[25] Zhou Z, Zhang C, Xu C, Xiong F, Zhang Y, et al. Energy-Efficient Industrial Internet of UAVs for Power Line Inspection in Smart Grid. In Ieee Trans Ind Inform. IEEE. 2018;14:2705-2714.

[26] Zhang C, Xie Y, Bai H, Yu B, Li W, et al. A Survey on Federated Learning. Knowl-Based Syst. 2021;216:106775.

[27] Liu M, Xia Y, Zhao H, Guo L, Shi Z, et al. Federated Learning Technologies for 6G Industrial Internet of Things: From Requirements Vision to Challenges Opportunities. Elect Inf Technol. 2024;46:4335-4353.

[28] Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, et al. Towards Federated Learning at Scale: System Design. Proceedings of the machine learning and systems. 2019;1:374-388.

[29] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, et al. Advances and Open Problems in Federated Learning. Found Trends Mach Learn. ACMDL. 2021;14:1-210.

[30] Zhan Y, Zhang J, Hong Z, Wu L, Li P, et al. A Survey of Incentive Mechanism Design for Federated Learning. IEEE Trans Emerg Top Comput. IEEE. 2021;10:1035-1044.

[31] Zhao H, Sui M, Liu M, Zhu C, Xun W, et al. Are You Diligent Inefficient or Malicious? A Self-Safeguarding Incentive Mechanism for Large Scale-Federated Industrial Maintenance Based on Double Layer Reinforcement Learning. IEEE Internet Things J. 2024;11:19988-20001.

[32] Zeng R, Zeng C, Wang X, Li B, Chu X. A Comprehensive Survey of Incentive Mechanism for Federated Learning. 2021. ArXiv preprint: https://arxiv.org/pdf/2106.15406.

[33] Asad M, Moustafa A, Rabhi FA, Aslam M. Thf: 3-Way Hierarchical Framework for Efficient Client Selection and Resource Management in Federated Learning. IEEE Internet Things J. 2021;9:11085-11097.

[34] Qu Z, Duan R, Chen L, Xu J, Lu Z, et al. Context-Aware Online Client Selection for Hierarchical Federated Learning. IEEE Trans Parallel Distrib Syst. 2022;33:4353-4367.

[35] Wang Y, Su Z, Luan TH, Li R, Zhang K. Federated Learning With Fair Incentives and Robust Aggregation for Uav-Aided Crowdsensing. IEEE Trans Netw Sci Eng. 2021;9:3179-3196.

[36] Chen C, Gong S, Zhang W, Zheng Y, Kiat YC. DRL-Based Contract Incentive for Wireless-Powered and Uav-Assisted Backscattering MEC System. IEEE Trans Cloud Comput. 2024;12:264-276.

[37] Xie L, Su Z, Chen N, Xu Q. Secure Data Sharing in UAV-Assisted Crowdsensing: Integration of Blockchain and Reputation Incentive. In: Proceedings of the 2021 IEEE global communications conference. GLOBECOM. Madrid. 2021:1-6.

[38] Kang J, Xiong Z, Niyato D, Xie S, Zhang J. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. IEEE Internet Things J. 2019;6:10700-10714.

[39] Wang Y, Sui M, Xia T, Liu M, Yang J, et al. Energy-Efficient Federated Learning-Driven Intelligent Traffic Monitoring: Bayesian Prediction and Incentive Mechanism Design. Electronics. 2025;14:1891.

[40] Chabrol M, Sarramia D, Tchernev N. Urban Traffic Systems Modelling Methodology. Int J Prod Econ. 2006;99:156-176.

[41] Abbasi M, Shahraki A, Taherkordi A. Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. Comput Commun. 2021;170:19-41.

[42] Liu M, Lin W, Wang Q, Gui G. Survey on Data Heterogeneity Problems and Personalization-Based Solutions of Federated Learning in Internet of Vehicles. J Commun. 2024;45:207-224.

[43] Zhao Y, Li M, Lai L, Suda N, Civin D, et al. Federated Learning With Non-IID Data. 2018. ArXiv preprint: https://arxiv.org/pdf/1806.00582.

[44] Neuenschwander B, Capkun-Niggli G, Branson M, Spiegelhalter DJ. Summarizing Historical Information on Controls in Clinical Trials. Clin Trials. 2010;7:5-18.

[45] Spiegelhalter DJ, Abrams KR, Myles JP. Bayesian Approaches to Clinical Trials and Health-Care Evaluation. Chichester UK: John Wiley & Sons. 2004.

[46] Wu Q, Zhang R. Common Throughput Maximization in Uav-Enabled Ofdma Systems With Delay Consideration. IEEE Trans Commun. 2018;66:6614-6627.

[47] Albaseer A, Abdallah M, Al-Fuqaha A, Erbad A. Client Selection Approach in Support of Clustered Federated Learning Over Wireless Edge Networks. In: Proceedings of the 2021 IEEE global communications conference. GLOBECOM. Madrid. 2021:1-6.

[48] Tang M, Ning X, Wang Y, Sun J, Wang Y, et al. Fedcor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. New Orleans. USA. 2022;10:10102-10111.

[49] Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. 2017. ArXiv preprint: https://arxiv.org/pdf/1708.07747v1.

[50] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-Efficient Learning of Deep Networks From Decentralized Data. In: Proceedings of the artificial intelligence and statistics. Fort Lauderdale. FL: PMLR. 2017:1273-1282.