HierFedKD-Traffic: Hierarchical Federated Knowledge Distillation for Wireless Traffic Prediction

Ons Aouedi and Symeon Chatzinotas
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg,
29 Avenue J.F. Kennedy, L-1855 Luxembourg
Corresponding author: ons.aouedi@uni.lu

***Abstract*—Accurate wireless traffic forecasting is essential for proactive resource allocation in 5G/6G networks, yet conventional centralized training over spatially distributed data raises concerns regarding privacy, latency, and backhaul capacity. Federated learning (FL) alleviates these issues by retaining data at the network edge; however, its performance degrades under the highly non-IID traffic distributions characteristic of heterogeneous cellular deployments. We present HIERFEDKD-TRAFFIC, a hierarchical FL framework purpose-built for wireless traffic prediction. HIERFEDKD-TRAFFIC introduces three tightly integrated mechanisms: (i) a base–core–student model decomposition aligned with the cloud–edge–base-station (BS) hierarchy, confining heavy computation to the edge while BSs retain only a lightweight predictor; (ii) a geo- and traffic-aware clustering that maps BSs to edge servers by jointly exploiting geographic proximity and diurnal load similarity, yielding coherent edge groups that stabilize teacher learning; and (iii) an adaptive knowledge distillation (KD) protocol that transfers edge-teacher knowledge to the on-BS student only when the teacher demonstrably outperforms it. On real call detail record (CDR) data from the Telecom Italia Big Data Challenge (Milano, Trento), HIERFEDKD-TRAFFIC reduces RMSE by 2.3–3.1% over FedAvg, FedProx, and HierFL, while simultaneously cutting per-round BS→edge communication by ≈22%, edge→cloud communication by ≈56%, and BS-side computation by ≈22%.**

***Index Terms*—wireless traffic prediction, federated learning, knowledge distillation, hierarchical learning**

I. INTRODUCTION

Accurate wireless traffic prediction enables anticipatory resource allocation, capacity planning, and anomaly detection in mobile networks [1], [2]. Conventional approaches centralize raw traces from distributed base stations (BSs) at a single server [3], incurring substantial backhaul overhead and raising regulatory concerns. Federated learning (FL) mitigates these issues by exchanging only model updates [4], [5]; however, three characteristics of wireless traffic render naïve FL insufficient.

(C1) Strong spatial non-IID. Traffic profiles vary significantly across cell types: residential cells peak in evenings, business-district cells during working hours, and event-venue cells exhibit irregular bursts. This spatial heterogeneity degrades flat FedAvg convergence [4].

(C2) Inherent network hierarchy. Operational mobile networks already possess a cloud–edge–BS continuum; flat FL ignores this topology, forcing every BS to communicate with a single central server and missing the opportunity for regional aggregation and cross-tier collaboration.

(C3) Asymmetric compute budgets. Capturing both short-term dynamics and long-range diurnal periodicity requires expressive temporal models, yet BSs are resource-constrained. Edge servers, by contrast, have ample compute and can host more powerful models.

These domain-specific challenges motivate HIERFEDKD-TRAFFIC, a hierarchical FL framework that addresses (C1)–(C3) jointly. Our contributions are:

- A *base–core–student* model decomposition (Sec. III-A) that keeps BS inference lightweight ($< 40\%$ of a full LSTM) while confining a powerful bi-LSTM teacher to the edge, directly addressing (C3).
- A *geo- and traffic-aware clustering* (Sec. III-B) that groups BSs by both location and diurnal load shape, forming coherent edge groups that improve teacher quality and stabilize distillation under non-IID traffic (C1).
- An *adaptive KD protocol* (Sec. III-C) that modulates distillation intensity based on the teacher–student performance gap, avoiding harmful over-distillation once the student matches the teacher.
- Experimental validation on two real-world CDR datasets, showing that HIERFEDKD-TRAFFIC simultaneously improves accuracy and reduces communication and computation overhead (Sec. IV).

The remainder of the paper is organized as follows. Section II reviews related work. Section III presents HIERFEDKD-TRAFFIC. Section IV reports experimental results. Section V concludes.

ISBN 978-3-903176-82-9

II. Related Work

FL for traffic prediction. FedDA [3] introduced dual attention to re-weight local updates under heterogeneous traffic. Subsequent studies confirmed that FL can approach centralized accuracy while preserving privacy [6]–[8]. However, these works adopt flat single-tier architectures and deploy monolithic predictors at BSs, disregarding the compute asymmetry between BSs and edges and missing the opportunity for cross-tier knowledge transfer.

Hierarchical FL. HierFL [9] introduces client–edge–cloud aggregation but uses *homogeneous* models across all tiers, so BSs still bear the full computational load. FLaTEC [10] splits a single model across thing–edge–cloud tiers via split learning, which requires cross-tier backpropagation and tight synchronization. Neither approach addresses the non-IID challenge specific to wireless traffic through traffic-aware client grouping.

Unlike HierFL—which aggregates a single heavy model at both BSs and edges—and unlike FLaTEC—which partitions one model and back-propagates gradients across tiers—HierFedKD-Traffic deploys heterogeneous, tier-matched, and independently trained models: a lightweight dual-LSTM base and a compact MLP student at BSs, and a high-capacity bi-LSTM teacher at edges. These models interact exclusively through federated aggregation and KD; no cross-tier gradient flow is required. Moreover, while prior hierarchical FL works assign clients to edges randomly or purely by proximity, HierFedKD-Traffic clusters BSs by jointly considering geography *and* diurnal traffic similarity, a design choice grounded in the periodicity and spatial heterogeneity specific to wireless traffic. To our knowledge, no existing work simultaneously addresses tier-aligned heterogeneous model capacity, traffic-aware clustering, and adaptive distillation for wireless time-series prediction.

III. Proposed HierFedKD-Traffic Framework

As shown in Fig. 1, HierFedKD-Traffic operates across the cloud–edge–BS continuum with a tier-aligned model decomposition, intelligent clustering, and an adaptive training protocol.

A. Model Decomposition

The predictor is decomposed into three components matched to the resources of each tier.

Base B_{θ_B} (BSs). A dual-LSTM feature extractor processes two complementary temporal windows: a *closeness* window $\mathbf{X}^c \in \mathbb{R}^{\tau_c \times d_{\text{in}}}$ capturing recent dynamics, and a *periodic* window $\mathbf{X}^p \in \mathbb{R}^{\tau_p \times d_{\text{in}}}$ sampling the same time-of-day over previous days. Each is processed by a single-layer LSTM:

$$\mathbf{h}_{k,c} = \mathrm{LSTM}_c(\mathbf{X}_k^c), \quad \mathbf{h}_{k,p} = \mathrm{LSTM}_p(\mathbf{X}_k^p). \tag{1}$$

The two views are fused by a learned element-wise gate and refined through a residual MLP with layer normalization:

$$\mathbf{g}_k = \sigma\big(\mathbf{W}_g[\mathbf{h}_{k,c}; \mathbf{h}_{k,p}] + \mathbf{b}_g\big), \tag{2}$$

$$\mathbf{h}_k^{\text{fused}} = \mathbf{g}_k \odot \mathbf{h}_{k,c} + (\mathbf{1} - \mathbf{g}_k) \odot \mathbf{h}_{k,p}, \tag{3}$$

$$\mathbf{h}_k = \mathrm{LN}\big(\mathbf{h}_k^{\text{fused}} + \mathbf{W}_2\, \phi(\mathbf{W}_1\, \mathbf{h}_k^{\text{fused}})\big), \tag{4}$$

where $\odot$ is the Hadamard product and ϕ denotes a GELU activation.

Core C_{θ_C} (edges, teacher). A deeper encoder composed of an input projection, a multi-layer bidirectional LSTM with self-attention, and an output MLP:

$$\mathbf{z} = \phi(\mathbf{W}_{\text{in}}^C \mathbf{h}_k + \mathbf{b}_{\text{in}}^C), \quad \tilde{\mathbf{z}} = \mathrm{BiLSTM}(\mathbf{z}), \tag{5}$$

$$\hat{y}_k^{(C)} = f_{\text{out}}(\mathrm{Attention}(\tilde{\mathbf{z}})). \tag{6}$$

The core operates on features $\mathbf{h}_k$ uploaded by BSs and executes at the edge tier, directly addressing the BS–edge compute asymmetry (C3).

Student S_{θ_S} (BSs). A compact MLP comprising an input projection ($\mathrm{Linear} \to \mathrm{LN} \to \mathrm{GELU} \to \mathrm{Dropout}$), a residual block (two $\mathrm{LN} \to \mathrm{GELU}$ layers with skip connection), and an output projection. The hidden dimension is $d_s = 64$, keeping the total parameter count below 40% of the full LSTM baseline. At inference, each BS runs only $(B_{\theta_B}, S_{\theta_S})$; the core remains at the edge.

B. Geo- and Traffic-Aware Clustering

Prior hierarchical FL assigns BSs to edges either randomly [10] or by geography alone. HierFedKD-Traffic exploits the domain knowledge that nearby cells with similar diurnal load profiles benefit most from a shared teacher. Each BS k is characterized by coordinates $(\mathrm{lng}_k, \mathrm{lat}_k)$, a normalized 24-hour traffic profile $\mathbf{t}_k \in \mathbb{R}^{24}$, and a variability descriptor cv_k (coefficient of variation). We build augmented feature vectors:

$$\mathbf{u}_k = [\,\gamma\,\mathrm{lng}_k,\ \gamma\,\mathrm{lat}_k,\ \mathbf{t}_k,\ \beta\,\mathrm{cv}_k\,], \tag{7}$$

where the weights γ and β balance the spatial and temporal scales. K-means then yields E clusters $\{\mathcal{C}_e\}_{e=1}^E$, each assigned to one edge server. As a result, each edge teacher is trained on a coherent subregion of the network, which improves its predictive accuracy and stabilizes distillation, mitigating.

C. Round-Based Training Protocol

Training proceeds over T rounds with cluster-aware partial participation (a fraction ρ of BSs per cluster per round).

Step 1 – BS local training. Within each cluster $\mathcal{C}_e$, a fraction ρ of BSs is sampled. Each selected BS k initializes local copies (B_k, S_k) from the current global

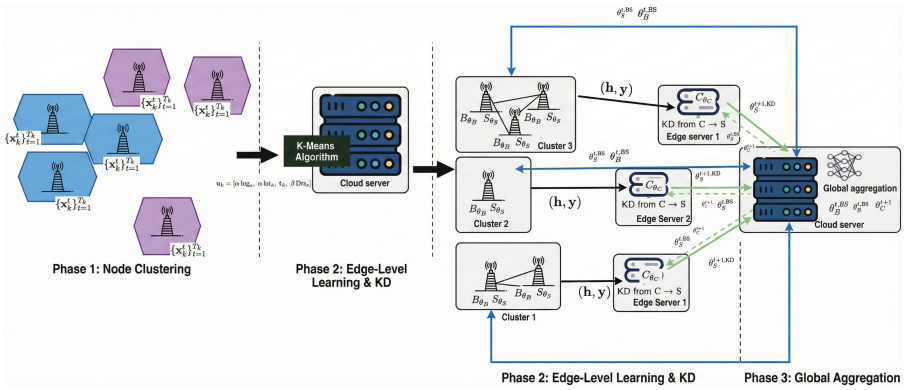

Fig. 1. Overview of HIERFEDKD-TRAFFIC. The cloud–edge–BS hierarchy hosts a tier-aligned model decomposition: a lightweight *base* B_{θ_B} and *student* S_{θ_S} run on every BS, while a high-capacity *core* teacher C_{θ_C} resides at each edge. BSs are grouped by joint geo- and traffic-aware clustering.

parameters and performs E_{loc} epochs of supervised training:

$$\mathcal{L}_k^{\text{sup}} = \frac{1}{|\mathcal{D}_k|} \sum_{(\mathbf{x},y)\in\mathcal{D}_k} \left\| S_{\theta_{S,k}}\big(B_{\theta_{B,k}}(\mathbf{x})\big) - y \right\|_2^2. \quad (8)$$

Step 2 – Feature upload. Each BS sub-samples a fraction p of its local data and extracts compact representations using the *global* (not locally updated) base model, which preserves cross-BS feature-space consistency:

$$\tilde{\mathcal{H}}_k = \big\{\big(B_{\theta_B^t}(\mathbf{x}),\, y\big) : (\mathbf{x}, y) \in \tilde{\mathcal{D}}_k\big\}, \quad |\tilde{\mathcal{D}}_k| \approx p\,|\mathcal{D}_k|. \quad (9)$$

The pairs are forwarded to the associated edge server, which maintains a rolling FIFO buffer $\mathcal{H}_e$.

Step 3 – Hierarchical aggregation. Base and student updates are aggregated via weighted FedAvg,

$$\theta_X^{t,\text{BS}} = \sum_{k\in\mathcal{K}_t} \frac{|\mathcal{D}_k|}{\sum_{j\in\mathcal{K}_t}|\mathcal{D}_j|}\, \theta_{X,k}^{t+1}, \quad X \in \{B, S\}, \quad (10)$$

implemented hierarchically: each edge first aggregates updates within its cluster, and the cloud then combines the edge-level summaries.

Step 4 – Periodic edge teacher update. After a warmup of t_{warm} rounds, every d_{edge} rounds each edge e refines its core on $\mathcal{H}_e$:

$$\mathcal{L}_e^{\text{core}} = \frac{1}{|\mathcal{H}_e|} \sum_{(\mathbf{h},y)\in\mathcal{H}_e} \left\| C_{\theta_{C,e}}(\mathbf{h}) - y \right\|_2^2. \quad (11)$$

The updated cores are aggregated as $\theta_C^{t+1} = \sum_e \frac{|\mathcal{H}_e|}{\sum_{e'}|\mathcal{H}_{e'}|}\, \theta_{C,e}^{t+1}$. Periodic (rather than per-round) synchronization reduces edge→cloud traffic.

Step 5 – Adaptive knowledge distillation. Let $\Delta_e^t = \text{RMSE}_{S,e}^t - \text{RMSE}_{C,e}^t$ denote the student–teacher gap at edge e, evaluated on the buffer $\mathcal{H}_e$. HIERFEDKD-TRAFFIC modulates KD intensity through three regimes:

- *Strong KD* ($\Delta_e^t > \delta$): $E_{\text{KD}} = 5$, $\alpha_{\text{soft}} = 0.8$, $\lambda_{\text{KD}} = 0.8$.
- *Mild KD* ($0 \le \Delta_e^t \le \delta$): $E_{\text{KD}} = 2$, $\alpha_{\text{soft}} = 0.5$, $\lambda_{\text{KD}} = 0.4$.
- *Skip* ($\Delta_e^t < 0$): $\lambda_{\text{KD}} = 0$ (the student already outperforms the teacher).

The KD objective combines a soft regression distillation term with a hard supervised term:

$$\mathcal{L}_e^{\text{KD}} = \alpha_{\text{soft}} \underbrace{\text{MSE}\big(S(\mathbf{h}),\, C(\mathbf{h})\big)}_{\mathcal{L}_{\text{soft}}} + (1-\alpha_{\text{soft}}) \underbrace{\text{MSE}\big(S(\mathbf{h}),\, y\big)}_{\mathcal{L}_{\text{hard}}}. \quad (12)$$

After E_{KD} epochs, the distilled student parameters from all edges are aggregated and blended with the BS-supervised update:

$$\theta_S^{t+1} = (1-\lambda_{\text{KD}})\, \theta_S^{t,\text{BS}} + \lambda_{\text{KD}}\, \theta_S^{t+1,\text{KD}}. \quad (13)$$

At inference, each BS evaluates only $(B_{\theta_B^T},\, S_{\theta_S^T})$: $\hat{y} = S_{\theta_S^T}\big(B_{\theta_B^T}(\mathbf{x})\big)$.

D. Communication and Computation Characteristics

Let $|w_{\text{full}}|$ denote the parameter count of the full LSTM baseline, $|w_{\text{BS}}| = |w_{\text{base}}| + |w_{\text{student}}|$ the on-device footprint of HIERFEDKD-TRAFFIC, and $|w_{\text{core}}|$ the edge-side teacher.

BS→edge communication. FedAvg and HierFL each transmit $\Theta(|w_{\text{full}}|)$ per selected BS per round. HIERFEDKD-TRAFFIC transmits $\Theta\big(|w_{\text{BS}}| + m_k(d_h + d_{\text{out}})\big)$, where $m_k \approx p|\mathcal{D}_k|$ feature–label pairs are uploaded. Communication is reduced whenever $|w_{\text{BS}}| + m_k(d_h + d_{\text{out}}) < |w_{\text{full}}|$, a condition readily satisfied for $|w_{\text{BS}}| < |w_{\text{full}}|$ and small p.

Edge→cloud communication. HIERFEDKD-TRAFFIC forwards aggregated base+student summaries every round and synchronizes the core teacher only every d_{edge} rounds, yielding a per-round cost of $\mathcal{O}\big(E \cdot |w_{\text{BS}}| + \frac{E}{d_{\text{edge}}}|w_{\text{core}}|\big)$, substantially below FedAvg's $\mathcal{O}(\rho K \cdot |w_{\text{full}}|)$.

BS-side FLOPs. Each BS trains only (B, S) with per-sample FLOPs $F_{\text{BS}} < F_{\text{full}}$. The reduction ratio is

$F_{\text{BS}}/F_{\text{full}} \approx 0.78$ in our architecture. Edge-side teacher training (F_{core} per sample, every d_{edge} rounds) is offloaded to resource-rich edge servers.

IV. Experimental Evaluation

A. Setup

Datasets. We use the Telecom Italia Big Data Challenge dataset [11], which contains Internet traffic from Milano (150 cells) and Trento (100 cells), resampled to hourly resolution. The first seven weeks serve as training data and the last week is held out for testing. Each cell corresponds to one FL client.

Baselines. We compare HierFedKD-Traffic against the following baselines:

- **LocalOnly:** each BS trains its own LSTM predictor on local data only, without any collaboration.
- **FedAvg** [4]: standard federated averaging of a single global LSTM predictor across BSs under a cloud coordinator.
- **FedProx** [12]: a robust FL variant that adds a proximal regularizer to stabilize training under non-IID data.
- **HierFL** [9]: client–edge–cloud hierarchical FL with two-stage aggregation (device→edge and edge→cloud), without knowledge distillation.

All FL methods share $\rho = 0.1$, $E_{\text{loc}} = 5$, and $T = 150$ communication rounds; reported results are averaged over five random seeds.

Implementation details. The experiments are implemented in PyTorch and run on a CPU-based simulation. Communication cost is estimated from the size of transmitted objects (model parameters and feature–label representations) assuming FP32 (4 bytes per scalar). Computation cost is measured as the total BS-side floating-point operations (FLOPs) accumulated over the $T = 150$ rounds. HierFedKD-Traffic uses $E = 4$ clusters, $d_{\text{edge}} = 5$, $p = 0.2$, and $\delta = 0.003$.

B. Prediction Accuracy

Table I summarizes the results. LocalOnly performs worst on both datasets, confirming that cross-cell collaboration is essential under data scarcity and non-IID conditions. FedProx provides no consistent gain over FedAvg, likely because the dominant challenge in this setting is structural non-IID heterogeneity rather than gradient-level client drift. HierFL improves over FedAvg on Milano (0.4528 vs. 0.4555) but *not* on Trento (0.4597 vs. 0.4572), showing that hierarchical aggregation alone—without cross-tier knowledge transfer—does not consistently help under strongly heterogeneous traffic.

HierFedKD-Traffic achieves the lowest error on both datasets, with RMSE of 0.4424 on Milano and 0.4456 on Trento, corresponding to reductions of 2.9% and 2.5% over FedAvg and 2.3% and 3.1% over HierFL, respectively. Importantly, these gains are obtained

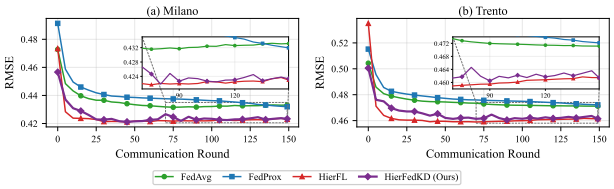

Fig. 2. Test RMSE vs. communication round on (a) Milano and (b) Trento.

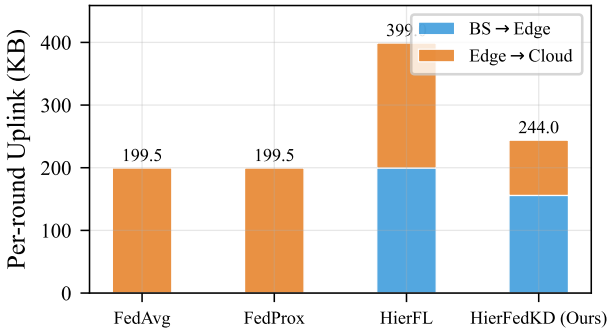

Fig. 3. Per-round uplink communication (KB), with BS→edge and edge→cloud shown separately.

despite deploying a BS-side predictor with $< 40\%$ of the full-model parameters.

Fig. 2 further shows the convergence behavior. HierFedKD-Traffic converges faster and remains consistently below all baselines throughout training on both datasets. The zoomed insets on the converged region (rounds 70–150) confirm that the accuracy advantage is maintained at steady state and is not a transient effect. The learning curve stays smooth despite periodic edge teacher updates, indicating that the hierarchical teacher–student interactions do not destabilize training.

C. Communication and Computation Efficiency

Fig. 3 reports per-round uplink message sizes broken down by link. FedAvg and FedProx are flat baselines: every selected BS transmits a full-model update of $\approx$199.5 KB directly to the cloud (no edge tier). HierFL introduces edge aggregation but transmits the same full model on both links, doubling the per-round payload. HierFedKD-Traffic reduces BS→edge communication by 22% (155.8 vs. 199.5 KB) by uploading only the compact base+student parameters together with a sampled set of feature–label pairs. On the edge→cloud link, it achieves a 56% reduction (88.2 vs. 199.5 KB) since only aggregated compact updates are forwarded and the core teacher is synchronized periodically (every $d_{\text{edge}} = 5$ rounds).

Fig. 4 reports total BS-side FLOPs over the $T = 150$ rounds. FedAvg, FedProx, and HierFL all incur high BS-side computation because each BS trains the full LSTM predictor; FedProx is slightly higher than FedAvg owing to the proximal-term overhead. HierFedKD-

TABLE I
PREDICTION PERFORMANCE ON THE TELECOM ITALIA BIG DATA CHALLENGE DATASET (LOWER IS BETTER; MEAN OVER 5 SEEDS).

	Milano (150 cells)			Trento (100 cells)		
Method	RMSE	MAE	NRMSE	RMSE	MAE	NRMSE
LocalOnly	0.4750	0.4130	0.9500	0.4812	0.4184	0.9624
FedAvg [4]	0.4555	0.3961	0.9110	0.4572	0.3976	0.9144
FedProx [12]	0.4590	0.3991	0.9180	0.4600	0.4000	0.9200
HierFL [9]	0.4528	0.3937	0.9056	0.4597	0.3997	0.9194
HIERFEDKD-TRAFFIC (Ours)	**0.4424**	**0.3847**	**0.8848**	**0.4456**	**0.3875**	**0.8912**

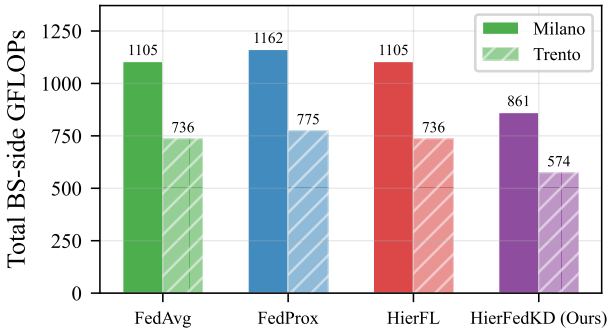

Fig. 4. Total BS-side GFLOPs accumulated over $T = 150$ communication rounds.

TRAFFIC cuts BS-side FLOPs by ≈22% relative to FedAvg/HierFL (861 vs. 1 105 GFLOPs on Milano; 574 vs. 736 GFLOPs on Trento), since each BS trains only the lightweight base+student.

Edge-side trade-off. The BS-side savings come at the cost of additional edge-side work, where edges train the high-capacity core teacher and run KD. In our setup, this adds ≈180 GFLOPs per edge over the $T = 150$ rounds for teacher training and KD combined—modest relative to the compute capacity of operational edge servers, which are typically provisioned with multi-core CPUs and, increasingly, GPUs. Periodic (every $d_{\text{edge}} = 5$ rounds) teacher updates further bound this overhead. Overall, HIERFEDKD-TRAFFIC reallocates computational load from resource-limited BSs to resource-rich edge nodes, in line with the practical conditions of mobile networks, where edge capacity is plentiful while BS resources are scarce.

D. Ablation Study

To understand which components drive the observed improvements, we view the baselines as successive ablations of HIERFEDKD-TRAFFIC. Table II reports the relative RMSE degradation $\Delta = \left(\text{RMSE}_{\text{variant}} - \text{RMSE}_{\text{Ours}}\right)/\text{RMSE}_{\text{Ours}} \times 100\%$ when each component is removed.

(i) Collaboration is essential. Removing all cross-BS collaboration (LocalOnly) causes the largest degradation (7–8%), confirming that FL-based knowledge sharing is critical under data scarcity and non-IID conditions.

TABLE II
ABLATION STUDY: RELATIVE RMSE DEGRADATION (Δ%) WHEN COMPONENTS ARE REMOVED FROM HIERFEDKD-TRAFFIC.

Variant	**Removed**	**Milano**	**Trento**
HIERFEDKD-TRAFFIC (Ours)	—	—	—
HierFL (w/o KD)	Edge-side KD	+2.35%	+3.16%
FedAvg (w/o hier.)	Hierarchy + KD	+2.96%	+2.60%
LocalOnly	All collaboration	+7.37%	+7.99%

(ii) Hierarchy alone is insufficient. HierFL improves over FedAvg on Milano but degrades on Trento. Hierarchical aggregation reorganizes communication but does not address non-IID content; without cross-tier knowledge transfer, the benefit is dataset-dependent and unreliable.

(iii) KD is the primary accuracy driver. The HierFL→HIERFEDKD-TRAFFIC gap (2.3–3.1%) exceeds the FedAvg→HierFL gap (<1%), isolating edge-side KD as the dominant contributor. The core teacher—trained on pooled features from a coherent cluster—learns a smoother, more generalizable mapping than any individual BS model, and the student inherits this signal via distillation.

(iv) Clustering stabilizes KD. Geo- and traffic-aware clustering groups cells with similar diurnal patterns at each edge, so the teacher sees a statistically coherent feature distribution. This improves teacher quality and makes the distilled knowledge more relevant.

(v) Adaptive gating prevents over-distillation. In later rounds the student can approach or surpass the teacher. Without gating, forcing continued KD degrades RMSE by up to 0.5% due to noise from a stale or weaker teacher; the gap-based mechanism avoids this by skipping KD when $\Delta_e^t < 0$.

Finally, each component targets a specific wireless-traffic challenge: the dual-LSTM base encodes diurnal periodicity (C1); the B/C/S decomposition matches tier-heterogeneous compute (C3); traffic-aware clustering mitigates spatial non-IID effects (C1); hierarchical aggregation exploits the network topology (C2); and adaptive KD transfers rich edge knowledge to BSs without inflating inference cost. This combination is not arbitrary: Table II shows that removing any single element degrades performance, and the accuracy gain of

HIERFEDKD-TRAFFIC over HierFL—which shares the hierarchy but lacks KD—confirms that the integration of edge-side distillation is the key differentiator.

E. Limitations

Our evaluation relies on simulation with FP32-based cost estimates and does not capture real-world network effects such as latency jitter, packet loss, or straggler nodes. Clustering is performed once using historical statistics; production deployments with evolving traffic would benefit from periodic online re-clustering. The architecture is LSTM-centric; exploring Transformer or state-space backbones is a natural extension. Finally, robustness to adversarial participants and noisy feature uploads remains to be investigated.

V. CONCLUSION

We presented HIERFEDKD-TRAFFIC, a three-tier hierarchical FL framework for wireless traffic prediction that integrates tier-aligned model decomposition, traffic-aware clustering, and adaptive edge-side knowledge distillation. Experiments on two real-world CDR datasets demonstrate consistent improvements: 2.3–3.1% RMSE reduction over representative FL baselines, 22% lower BS$\rightarrow$edge communication, 56% lower edge$\rightarrow$cloud communication, and 22% less BS-side computation. The ablation study confirms that edge-side KD is the primary accuracy driver, that traffic-aware clustering stabilizes distillation, and that each design choice is grounded in a specific characteristic of wireless traffic prediction. Future work will explore dynamic clustering, alternative backbones (e.g., Transformers), and robustness extensions, including straggler handling and adversarial participants.

REFERENCES

[1] O. Aouedi, V. A. Le, K. Piamrat, and Y. Ji, "Deep learning on network traffic prediction: Recent advances, analysis, and future directions," *ACM Computing Surveys*, vol. 57, no. 6, pp. 1–37, 2025.

[2] O. Aouedi, K. Piamrat, and B. Parrein, "Intelligent traffic management in next-generation networks," *Future internet*, vol. 14, no. 2, p. 44, 2022.

[3] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *IEEE INFOCOM 2021-IEEE conference on computer communications*. IEEE, 2021, pp. 1–10.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[5] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, vol. 195, pp. 346–361, 2022.

[6] S. Behera, S. K. Panda, T. Panayiotou, and G. Ellinas, "Federated learning for network traffic prediction," in *2024 IFIP Networking Conference (IFIP Networking)*. IEEE, 2024, pp. 781–785.

[7] N. Pavlidis, V. Perifanis, S. F. Yilmaz, F. Wilhelmi, M. Miozzo, P. S. Efraimidis, R.-A. Koutsiamanis, P. Mulinka, and P. Dini, "Federated learning in mobile networks: A comprehensive case study on traffic forecasting," *IEEE Transactions on Sustainable Computing*, 2024.

[8] S. K. Panda, B. Palit, and S. Behera, "Fednet: Federated learning for proactive traffic management and network capacity planning," *arXiv preprint arXiv:2511.06797*, 2025.

[9] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.

[10] J. Haga, Y. Tanimura, T. T. Nguyen *et al.*, "Flatec: An efficient federated learning scheme across the thing-edge-cloud environment," *Future Generation Computer Systems*, p. 108073, 2025.

[11] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of milan and the province of trentino," *Scientific data*, vol. 2, no. 1, pp. 1–15, 2015.

[12] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.