International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Federated Demand Forecasting for Retail Supply Chains

When Does Sharing Gradients Pay Off? An Empirical Study on Walmart M5 Data under Item-Level Differential Privacy

Jatin Singh[1], Rahul Khurana[2]

Abstract: ***Retailers, distributors and store operators all forecast demand for the same products, but almost none of them will hand their sales history to one another. Federated learning offers a way around that stalemate: each party trains on its own data and shares only model updates. The question we care about is a practical one. Does federation actually make store-level forecasts better, and does it still help once the updates are protected with differential privacy? We study this on the public M5 dataset (30,490 SKU-store series from ten Walmart stores in three US states, 1,969 days), treating every store as a separate data silo. We compare local-only training, a pooled centralized model, FedAvg, FedAvg with local fine-tuning, and a DP-FedSGD variant that guarantees item-level (ε, δ)-differential privacy for each store's catalogue, with privacy computed exactly using Gaussian differential privacy. The answer depends mostly on how much history a store has. With two years of training windows, federation is close to a wash at the SKU level (FedAvg with fine-tuning scores 0.895 WRMSSE against 0.899 for local training) and its 4.2% reduction in store-total error is within seed noise. With only two weeks of windows, local models fall apart and federation recovers much of the loss (0.939 vs 0.971), yet a plain 28-day moving average (0.922) is still hard to beat. When one newly opened store joins nine mature ones, fine-tuning the federated model lowers that store's error by 7.5% on average over local training, and by 7.1% with ε = 2 privacy. We also report where the approach breaks: near-zero sales on Christmas Day, erratic SKUs, and private training when stores drop out of rounds. The full set of experiments runs in under half an hour on two CPU cores, and every number in this paper comes from our own runs.***

Keywords: federated learning; retail demand forecasting; differential privacy; cross-silo learning; SKU-level forecasting; M5 dataset

1. Introduction

Anyone who has worked on replenishment knows the pattern. The store sees the demand, the distribution center sees the orders, and the manufacturer sees a smoothed, delayed and often distorted version of both. Lee, Padmanabhan and Whang called the result the bullwhip effect: small swings at the shelf become large swings upstream, and one of the causes they identified is that each tier forecasts from the orders it receives rather than from end demand [10]. The obvious fix is to share point-of-sale data. The obvious problem is that sales history is commercially sensitive. A supplier who sees a retailer's SKU-level sales can infer promotions, margins and assortment plans.

Federated learning (FL) was designed for exactly this kind of standoff [1, 2]. Each participant keeps its raw data and sends model updates to an aggregator, which combines them into a shared model. Updates can still leak information, so FL is usually paired with differential privacy (DP), which bounds how much any single record can change what leaves the participant [3, 4]. In practice, two things are rarely checked. First, whether the shared model is actually better than what each participant could train alone, since retail stores differ in size, customer mix and local events. Second, how much accuracy is lost once the privacy guarantee is formal rather than aspirational.

This paper tries to answer those two questions with real data rather than a simulator. We use the M5 competition dataset [7], split it into ten store silos, and run every method under the same model, features and training budget. Our contributions are:

- **A reproducible cross-silo benchmark** for federated SKU-level demand forecasting on M5, with local, centralized, FedAvg, fine-tuned and differentially private variants trained under matched budgets.
- **An item-level DP design for retail** in which the protected unit is a whole SKU's sales history inside a store, with exact privacy accounting through Gaussian differential privacy [5].
- **A data-scarcity analysis** showing that the value of federation is concentrated in stores with short histories, and quantifying how far it closes the gap to a pooled model.
- **An edge-case study** covering new-store cold start, a Christmas window, intermittent demand classes and partial participation, including cases where federation does not help.

2. Background and Related Work

Federated learning: FedAvg [1] alternates between local training on each client and weighted averaging of the resulting models at a server. Much of the early FL work targeted the cross-device setting, with millions of phones and tiny local datasets. Supply chains are the other extreme: a handful of organizations ("silos"), each holding a lot of data, participating in every round. Kairouz et al. [2] survey both settings. Reddi et al. [6] showed that using an adaptive optimizer such as Adam at the server often helps when client data are non-identical, which we adopt for our private variant.

Differential privacy: DP-SGD [3] clips each example's gradient to a fixed norm and adds Gaussian noise before the update is used. McMahan et al. [4] moved the same idea to FL and changed the protected unit from a single example to a whole user. We borrow that idea and protect a whole SKU. For accounting we use Gaussian differential privacy (GDP) [5], under which T rounds of the Gaussian mechanism with noise multiplier σ compose to μ-GDP with $\mu = \sqrt{T} / \sigma$. This gives an exact conversion to (ε, δ) for full-batch training, with no looseness from composition bounds.

Retail forecasting and M5: The M5 accuracy competition asked participants to forecast 28 days of unit sales for 30,490 Walmart SKU-store series and scored them with weighted RMSSE [7]. RMSSE scales errors by the in-sample error of a naive forecast, an idea that goes back to Hyndman and Koehler's MASE [8]. Retail series at this level are mostly intermittent in the sense of Syntetos, Boylan and Croston [9], which makes them hard for any model.

FL for demand forecasting: Earlier studies show the idea is promising. Wang et al. [11] proposed a vertical FL framework that combines features held by different platforms (social media, e-commerce and retailers) for supply-chain demand forecasting. Long et al. [12] cluster heterogeneous retailers into groups using DP-protected feature importance and train a Transformer per group, reporting large gains over local learning and FedAvg. What we did not find is an evaluation on a widely used public retail benchmark that (i) reports formal (ε, δ) guarantees with a clearly stated protected unit, (ii) varies how much history each silo has, and (iii) checks the edge cases that planners care about. That is the gap this paper tries to fill.

3. Problem Setup

We have K = 10 stores. Store k holds daily unit sales $y_{k,i,t}$ for its catalogue of 3,049 SKUs, together with sell prices and the calendar. At a forecast origin t_0 each store needs forecasts for days t_0 through $t_0 + 27$ for every SKU, which is the M5 horizon. Stores are willing to cooperate on a shared model but not to reveal raw sales.

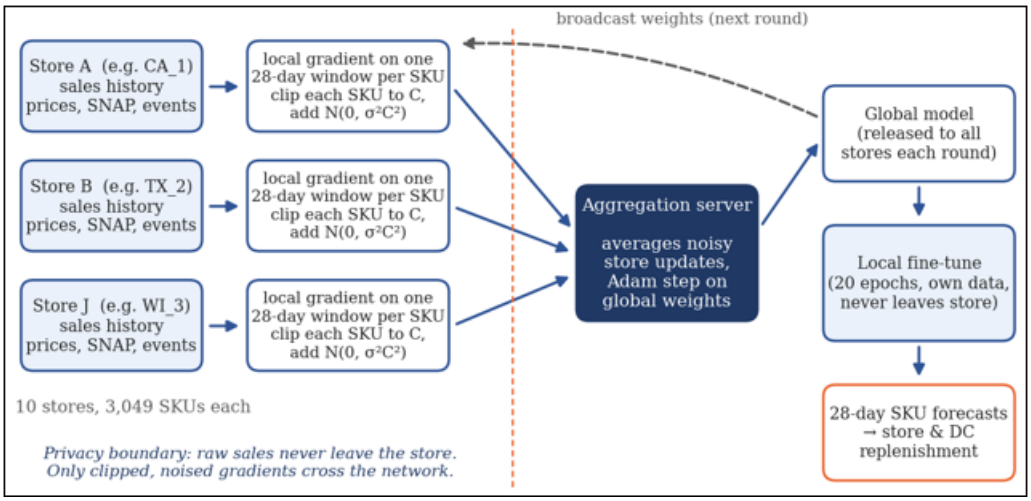


Figure 1: Training pipeline. Stores compute updates locally; in the private variant each SKU's contribution is clipped and noised before it leaves the store. The global model comes back each round, and the last step (fine-tuning) happens entirely inside the store

Threat model: We assume an honest-but-curious aggregator and honest-but-curious peer stores. Everything a store sends can be seen by the server, and the global model is broadcast to all stores. We do not consider stores that send malicious updates. The privacy goal is that no observer of the released updates can reliably tell whether a particular SKU's sales history was part of a given store's training data. We chose the SKU as the protected unit because that is where competitive value sits: a rival learning that one store sells a lot of a particular private-label product is a real business risk. Because each SKU contributes many days, protecting one day would be much weaker.

4. Method

4.1 Forecasting model

All learned methods share one small network so that differences come from how it is trained, not from architecture. The input for SKU i at origin t_0 is the last 56 days of sales divided by a scale m = 1 + mean (last 56 days), the log of m, event flags for each of the 28 target days, SNAP benefit-day flags for the store's state for each target day, a one-hot department code (7 departments), and the log ratio of the current sell price to its 56-day mean. That gives 121 inputs. A single hidden layer of 64 ReLU units maps them to 28 outputs, one per horizon day, which are multiplied back by m. The model has 9,628 parameters. A small model matters under DP, since the total noise grows with the parameter count. All training origins fall on Mondays, like the M5 test origin, so each output unit always corresponds to the same weekday and learns its day-of-week pattern. The loss is mean squared error on the scaled targets.

4.2 Training protocols

Every learned method trains for 100 rounds. In each round every store draws one random 28-day training window per SKU (3,049 windows per store) from the Monday origins allowed by the data-scarcity regime.

- **Local:** each store trains its own model with Adam (learning rate 10^{-3}, batch 256), one pass over its windows per round.
- **Centralized:** one model trained the same way on all stores' windows pooled together. It needs raw-data sharing, so we treat it as a reference point rather than a deployable option.
- **FedAvg** [1]: each store runs one local epoch of Adam from the current global weights, and the server averages the ten models weighted by their number of windows.

- **FedAvg + FT:** the final FedAvg model is fine-tuned by each store on its own data for 20 epochs. The fine-tuned model never leaves the store, so it costs nothing in privacy.
- **DP-FedSGD:** each store computes the gradient of every SKU's window, clips it to L2 norm C = 1, sums, adds Gaussian noise $N(0, \sigma^2C^2I)$, and divides by its public catalogue size. The server averages the noisy store gradients and takes one Adam step (learning rate 10^{-2}, chosen on the validation window). "+FT" again means local fine-tuning afterwards.

4.3 Privacy accounting

Because each SKU contributes exactly one clipped gradient per round, adding or removing one SKU changes a store's summed gradient by at most C. Each round is therefore a Gaussian mechanism with sensitivity C and noise multiplier σ, and 100 rounds compose to μ-GDP with $\mu = 10 / \sigma$ [5]. We convert μ to ε at $\delta = 10^{-5}$ (well below 1/3,049) by solving the exact GDP trade-off equation numerically. The noise multipliers that give ε = 1, 2, 4 and 8 are σ = 37.3, 19.9, 10.8 and 6.0. The guarantee holds for each store separately and covers everything the store sends. The global model and anything computed from it, including local fine-tuning, are post-processing and add no privacy cost.

5. Experimental Design

5.1 Data sources

Table 1 lists the data we used. We took the M5 files from the copy hosted in Nixtla's public m5-forecasts repository [13] and checked the structure directly: 30,490 series (the count reported in [7]), 10 stores (four in California, three in Texas, three in Wisconsin), 3,049 SKUs per store across 3 categories and 7 departments, and 1,969 days from 29 January 2011 to 19 June 2016. About 68% of all SKU-days are zero, counting days before a product was launched.

Table 1: Data sources and how each is used.

Source	Contents	Use in this study
M5 sales_train_evaluation.csv [7, 13]	Daily unit sales, 30,490 SKU-store series, days 1-1941	Training windows, model inputs, RMSSE scaling
M5 sales_test_evaluation.csv	Days 1942-1969 (23 May - 19 Jun 2016)	Held-out 28-day test horizon
M5 sell_prices.csv	Weekly price per SKU-store	Price-ratio feature; revenue weights for WRMSSE
M5 calendar.csv	Dates, holidays and events, SNAP days per state	Event and SNAP features for each target day
Derived silos	One client per store (CA_1-4, TX_1-3, WI_1-3)	Federated clients; no raw data crosses silos

Splits: The test origin is Monday 23 May 2016 (day 1942), and models only see data before it. On the previous 28-day window (origin 25 April 2016) we compared 30, 100 and 200 rounds, fine-tuning for 5 or 20 epochs, and DP server learning rates of 10^{-3}, 3×10^{-3} and 10^{-2}, then froze the choices (100 rounds, 20 epochs, 10^{-2}). We kept 100 rounds rather than 200 because more rounds cost more privacy budget and the validation gain was small. The Christmas experiment uses a separate origin, 14 December 2015.

5.2 Metrics

Our main metric is SKU-level WRMSSE, the level-12 metric of M5 [7]. For each series, RMSSE is the root mean squared forecast error divided by the root mean squared one-step naive error over that series' history (starting from its first sale). Series are weighted by their dollar sales in the 28 days before the origin. Because replenishment to a store is usually planned on totals, we also report store-level WRMSSE, computed the same way on the ten store-total series (M5 level 3). All learned methods are run with three random seeds and we report mean ± standard deviation.

5.3 Data-scarcity regimes and edge cases

To see when federation helps, we restrict how many Monday training origins each store may use: 104 weeks (full), 26, 8 and 2 weeks. At 2 weeks, each SKU has only two distinct training windows. Table 2 lists the edge cases we test on top of this.

Table 2: Edge-case scenarios and how each is simulated.

Scenario	What it stresses	How we simulate it
New store joins (cold start)	A silo with almost no history	One store limited to 2 weeks of windows, the other nine keep 104; repeated for each of the 10 stores
Christmas Day	Rare, extreme event not separable from other holidays in the features	Forecast origin 14 Dec 2015; 14 units sold chain-wide on 25 Dec vs about 36,000 the day before
Intermittent demand	Sparse, lumpy SKU series	Break test error down by Syntetos-Boylan-Croston class [9], computed on the last 365 days
Partial participation	Stores missing rounds (outages, stragglers)	Each store joins each round with probability 0.5 (8-week regime)
Heterogeneous stores	Non-identical demand across states and store sizes	Per-store store-level RMSSE, full-history regime

6. Results

6.1 How much history does a store need?

Table 3 and Figure 2 tell a fairly consistent story. With the full two years, every learned model beats both naive baselines at the SKU level, and the non-private learned approaches are within half a percent of one another (0.899 local, 0.895 federated with fine-tuning, 0.896 centralized). A store with this much data does not need anyone else's to fit a 28-day SKU forecaster. At the store-total level, fine-tuned FedAvg has 4.2% lower error than local training, but neither plain

FedAvg nor the centralized model beats local training there, and the gap is about one standard deviation across seeds. We would not build a business case on it. Note also that the seasonal naive forecast, which is poor at the SKU level, is a strong store-level baseline (0.718), because weekly seasonality dominates once SKUs are summed.

As history shrinks, local models degrade fastest. At 2 weeks, local training reaches 0.971, federation recovers to 0.939 and pooling all data reaches 0.934. Fine-tuned FedAvg therefore closes about 86% of the gap between local and centralized training, and plain FedAvg (0.935) closes nearly all of it. At this regime fine-tuning slightly hurts: two windows per SKU are not enough to adapt safely. The private model without fine-tuning (0.928) even lands below the centralized one, but its seed spread is ±0.015, so we read that as a tie. The more important caveat is the moving average. At 2 weeks it (0.922) beats every learned model. At 8 weeks, the federated and private models edge past it while the local model still does not. A planner with a few weeks of history should not expect a neural forecaster trained alone to beat a moving average; with federation, it becomes a close call.

Table 3: Test WRMSSE by training history (lower is better). SKU level unless marked; mean ± s.d. over 3 seeds.

Method	104 wk	26 wk	8 wk	2 wk	Store level, 104 wk	Store level, 2 wk
Seasonal naive (last week)	1.176	1.176	1.176	1.176	0.718	0.718
28-day moving average	0.922	0.922	0.922	0.922	0.913	0.913
Local only	0.899 ± 0.003	0.915 ± 0.003	0.927 ± 0.006	0.971 ± 0.004	0.660 ± 0.014	1.798 ± 0.051
FedAvg	0.898 ± 0.001	0.909 ± 0.006	0.916 ± 0.007	0.935 ± 0.004	0.703 ± 0.058	1.394 ± 0.128
FedAvg + fine-tune	0.895 ± 0.001	0.908 ± 0.006	0.915 ± 0.009	0.939 ± 0.003	0.633 ± 0.040	1.404 ± 0.118
DP-FedSGD, ε = 2	0.905 ± 0.002	0.908 ± 0.004	0.911 ± 0.005	0.928 ± 0.015	0.745 ± 0.059	1.126 ± 0.273
DP-FedSGD, ε = 2 + fine-tune	0.902 ± 0.002	0.908 ± 0.006	0.911 ± 0.007	0.932 ± 0.020	0.664 ± 0.030	1.205 ± 0.393
Centralized (pooled data)	0.896 ± 0.001	0.912 ± 0.001	0.920 ± 0.005	0.934 ± 0.011	0.679 ± 0.020	1.301 ± 0.177

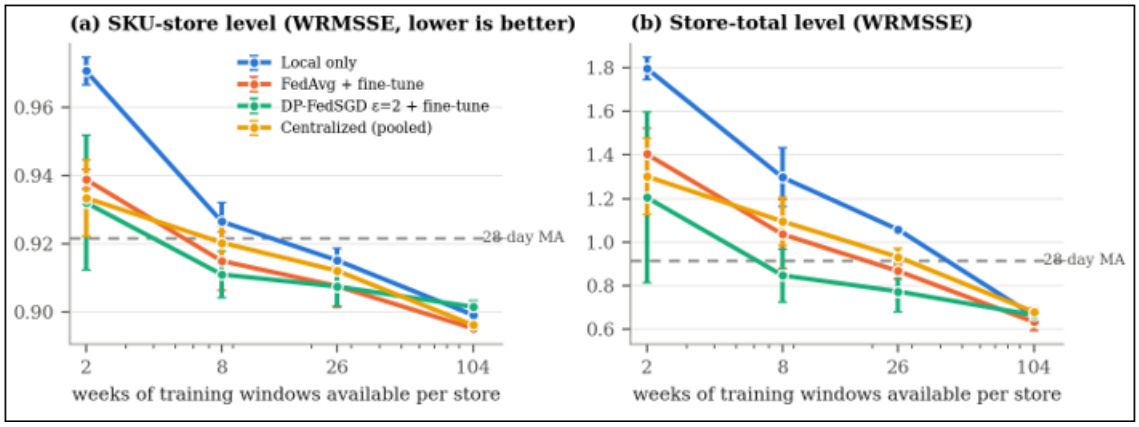


Figure 2: Test error against the amount of history per store. Error bars show ±1 s.d. over three seeds; the dashed grey line is the 28-day moving average, which uses no training at all

6.2 The price of privacy

Figure 3 shows the cost of formal privacy. With full history, ε = 8 and ε = 1 give SKU-level WRMSSE of 0.901 and 0.902 after fine-tuning, compared with 0.895 without DP, so the private model is about 0.7% worse at ε = 2. With full history, fine-tuning recovers part of the loss at every budget. In the 8-week regime the private model is roughly on par with, and at some budgets slightly better than, non-private FedAvg (0.911 at ε = 2 against 0.916). Part of this is likely the regularizing effect of clipping and noise, and part is that the private variant takes full-batch steps rather than local epochs, so the comparison mixes privacy with the optimizer. What we can say is that item-level privacy at single-digit ε does not wreck accuracy in either regime. The curve is also not monotone in ε in the scarce regime, fine-tuning slightly hurts there at ε = 4 and 8, and with three seeds most differences between budgets are within noise.

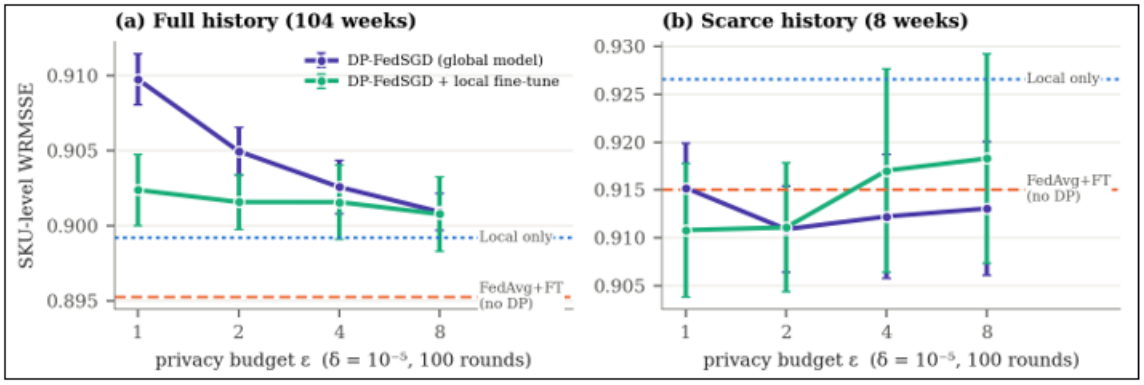


Figure 3: SKU-level test error of the private model as the privacy budget changes. Dotted and dashed lines show local training and non-private FedAvg + fine-tune for the same regime.

6.3 Edge Cases

A new store joins: Figure 4 shows the cold-start test, where one store has two weeks of windows and the other nine have two years. Fine-tuned FedAvg beats the new store's local model in 10 of 10 stores, with an average improvement of 7.5% (range 4.7% to 11.7%). The private version helps in 10 of 10 stores, by 7.1% on average. The fine-tuned federated model also beats the moving average in 10 of 10 stores, which the new store's own model never does. Plain FedAvg without fine-tuning is better still in 8 of 10 stores, which matches the regime results: a store with two weeks of data should use the shared model as it is. This is the clearest case for federation in our study, and the most realistic: new locations and new partners are exactly the ones with the least data.

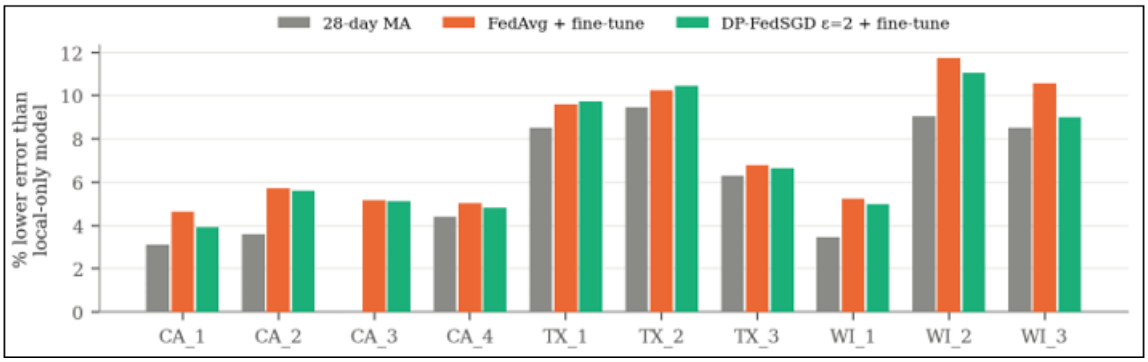


Figure 4: Cold start: SKU-level error reduction relative to the new store's local-only model, when that store has 2 weeks of training windows and the other nine have 104 weeks (one seed per store).

Christmas Day: In every year of M5, chain-wide sales on 25 December are between 11 and 20 units, against about 36,000 on a normal day, which is consistent with the stores being closed. Figure 5 shows what the models do with it. None of the models forecasts anything unusual for the 25th. The fine-tuned federated model predicts 31,792 units and the local models 34,835, roughly what they predict for a normal Friday. All learned models also under-forecast the pre-Christmas rush on 22-24 December. The reason is simple. Our features use one binary flag for every event day in the calendar (162 of them over five years), so Christmas looks the same as Super Bowl Sunday or Mother's Day, and the training windows contain only two Christmases. Federation cannot fix a feature that does not exist. In practice, known closures should be handled by a calendar rule outside the model. Across the whole window, the SKU-level WRMSSE was 0.886 for local, 0.881 for FedAvg + fine-tune, 0.891 for the private model and 0.901 for the moving average.

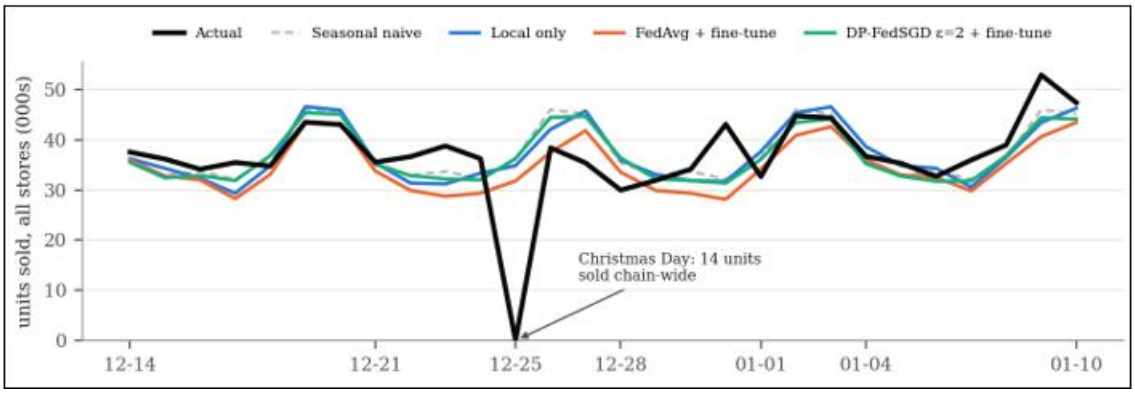


Figure 5: Chain-wide daily units, 14 Dec 2015 - 10 Jan 2016 (sum of SKU forecasts, seed 0, full history). No model anticipates the near-zero sales on Christmas Day, and all under-forecast the days just before it

Table 4: SKU-level WRMSSE by demand class (104 weeks, seed 0). Classes follow Syntetos, Boylan and Croston [9].

Demand class	Series	Revenue weight	28-day MA	Local	FedAvg + FT	DP ε=2 + FT
Smooth	2,947	32.30%	0.945	0.91	0.909	0.916
Erratic	686	4.20%	1.119	1.068	1.065	1.079
Intermittent	23,183	54.80%	0.886	0.871	0.869	0.876
Lumpy	3,665	8.70%	0.963	0.926	0.923	0.934

Intermittent demand: Table 4 breaks the full-history results down by demand class. Most series (23,183 of 30,490) are intermittent, but smooth series carry 32% of the revenue weight. Every learned model beats the moving average in every class. Errors are highest on erratic series for every method. Intermittent series score best, probably because RMSSE divides by a naive error that is itself large for sparse series. Fine-tuned FedAvg is better than local training on all four classes, and the private model is slightly worse than both on every class, but these differences are small next to the differences between classes. If a consortium wants better forecasts for erratic and lumpy SKUs, a model built for intermittency (for example a count distribution loss) will probably matter more than who shares updates with whom.

Stores missing rounds: When each store takes part in a round only half the time (8-week regime), FedAvg with fine-tuning scores 0.916 ± 0.011, against 0.915 with full participation, and the private model with fine-tuning 0.929 against 0.911. Non-private training barely notices. The private model does: its error rises by about 2%. That is what one would expect, since each round now averages the noise of about five stores instead of ten, so the noise in the global update grows by roughly √2 while the signal does not. Our accounting also still charges each store for all 100 rounds, which overstates its privacy loss when it skipped some. Private consortia should plan for dropouts, for example by raising the per-round participation requirement or scaling noise to the number of stores actually present.

Store heterogeneity: With full history, fine-tuned FedAvg has lower store-level error than local training in 6 of 10 stores. The stores where it does worse are CA_2, TX_1, TX_2, WI_1. With ten stores we could not find, and would not trust, a rule for which stores lose.

7. Discussion and Limitations

What this means for supply-chain partners: The case for federated forecasting is narrower than "everyone benefits". Large, established participants gain little, at the SKU level or in store totals. Participants with little history gain noticeably (7.5% on average for a newly joined store in our test), and they gain it without exposing their own data. That asymmetry has consequences for incentives. A mature store that contributes updates mainly helps newcomers, so a consortium may need a way to compensate data-rich members. The fact that fine-tuning is free under DP helps here, since every member can keep a personalized model on top of the shared one.

Threats to validity: Several limits apply. (1) All ten M5 stores belong to one retailer and share a catalogue and pricing policy. Real cross-company federation would involve different catalogues and much stronger heterogeneity, so our gains are probably an upper bound for that setting. (2) We use one small network and one feature set. Tree ensembles dominated M5 [7] and are harder to federate. (3) The comparison between DP-FedSGD and FedAvg mixes privacy with optimizer differences, as noted above. (4) Three seeds and one main test window are enough to see large effects but not small ones; the sub-half-percent differences at full history should be treated as ties. (5) DP protects SKU histories against observers of the updates, but not against a malicious server that deviates from the protocol, and we did not use secure aggregation. (6) We measure forecast accuracy, not inventory cost. Translating error reductions into safety stock and service levels requires lead times and cost data that M5 does not include. The natural next steps are a cross-retailer dataset, per-event features, count-based losses for intermittent SKUs, and accounting that credits stores for rounds they miss.

8. Conclusion

We set out to test whether federated learning improves demand forecasts for supply-chain partners who cannot share sales data, and whether it survives a formal privacy guarantee. On the M5 Walmart data, the answer is "yes, where data are scarce". A store with two years of history gains little at the SKU level from federation. A store with a few weeks of history gains clearly, especially when it joins partners who have more. Item-level differential privacy at ε between 1 and 8 costs under one percent of SKU-level accuracy at full history and little or nothing when data are scarce, although the private model is more sensitive to stores missing rounds. Some problems, such as holiday closures and erratic demand, are not solved by federation at all and need their own fixes. We hope the benchmark and the negative results are as useful as the positive ones.

References

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of AISTATS 2017*, PMLR 54, 1273-1282.
[2] Kairouz, P., McMahan, H. B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2), 1-210.
[3] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of ACM CCS 2016*, 308-318.
[4] McMahan, H. B., Ramage, D., Talwar, K., & Zhang, L. (2018). Learning differentially private recurrent language models. *ICLR 2018*. arXiv:1710.06963.
[5] Dong, J., Roth, A., & Su, W. J. (2022). Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B*, 84(1), 3-37. doi:10.1111/rssb.12454.
[6] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., & McMahan, H. B. (2021). Adaptive federated optimization. *ICLR 2021*. arXiv:2003.00295.
[7] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4), 1346-1364.
[8] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679-688.
[9] Syntetos, A. A., Boylan, J. E., & Croston, J. D. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society*, 56(5), 495-503.
[10] Lee, H. L., Padmanabhan, V., & Whang, S. (1997). Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4), 546-558.
[11] Wang, H., Xie, F., Duan, Q., Li, J., & Sharma, P. (2022). Federated learning for supply chain demand forecasting. *Mathematical Problems in Engineering*, 2022, 4109070.
[12] Long, Y., Xu, L., Zheng, G., & Brintrup, A. (2025). A bubble-cluster federated learning framework for privacy-preserving demand forecasting on heterogeneous retail data. arXiv:2503.12220.
[13] Nixtla. m5-forecasts repository, datasets/m5.zip (copy of the M5 competition data). https://github.com/Nixtla/m5-forecasts. Accessed September 2026.