Advances in Artificial Intelligence and Machine Learning; Research 6 (1) 4876-4890 Received 12-10-2025; Accepted 16-01-2026; Published 23-01-2026

Quantifying Cloud Cost Efficiency in Federated Learning: An Empirical Comparison with Centralized Training

Avijit Bose avijit.bose@mckvie.edu.in
Department of Computer Science & Engineering,
MCKV Institute of Engineering, Liluah,
Howrah-711204.

Pradyut Sarkar pradyut.sarkar@makautwb.ac.in
Department of Computer Science & Engineering,
MAKAUT, Kalyani, West Bengal,
India.

Sabyasachi Saha sabyasachi@technoexponent.com
Techno Exponent,
Miami, Floria,
USA.

Premanada Jana prema_jana@yahoo.com
Netaji Subhash Open University,
Kalyani, West Bengal, India -7112006.

Corresponding Author: Avijit Bose

Abstract

Machine learning has become an important component of data driven applications. Most large-scale machine learning jobs are now executed on cloud platforms. These deployments often face high bandwidth and storage cost issues which can overside computation expenses. The present study investigates whether FL (Federated Learning) can overcome such costs without compromising predictive performance. A controlled experiment was conducted using two clients and ten communication rounds on the California Housing dataset to compare a centralized configuration with a federated one. In the FL set up, only model parameters were exchanged thus eliminating repeated data transfer. Realistic cloud pricing models for data transfer, compute and temporary storage were applied to recorded execution statistics. Results showed that centralized training consumed nearly 5,000 × more network bandwidth than FL while producing similar accuracy (MSE = 0.542 vs 0.556; R^2 = 0.587 vs 0.576). Under limited-bandwidth conditions and when local data are available, FL substantially lowers operating expenses without compromising on model quality. The study also highlights deployment variables such as round budgeting, update sizing, and client selection that influence overall efficiency. These findings indicate that federated learning which was popular for its privacy preservation is a practical, economically feasible solution for cloud-based machine learning systems.

Citation: Avijit Bose, et al. Quantifying Cloud Cost Efficiency in Federated Learning: An Empirical Comparison with Centralized Training. **Advances in Artificial Intelligence and Machine Learning.** 2026;6(1):270. https://dx.doi.org/10.54364/AAIML.2026.61270

Keywords: Cloud cost modelling, Distributed machine learning, Edge–cloud collaboration, Federated learning, Resource allocation.

1. INTRODUCTION

Cloud-based machine learning now drives progress in finance, healthcare, and mobile analytics. The scope is broad, but so is the cost. Every single step in the process of training-moving the information, talking about the information, and storing the information safe-costs money and adds to the overall financial burden. Large volumes of data movement between the edge and the cloud can quickly become expensive [1]. Running machine-learning workloads in the cloud often consumes large amounts of bandwidth and storage, and those two items can easily exceed the actual cost of computation. Security and privacy rules make the picture even more complicated. Many organizations hesitate to move sensitive data to central servers because of legal and ethical constraints [2]. Cloud vendors charge for every gigabyte that leaves the data centre and for every second of processing time. For data owners, training models is therefore not a one-off investment but a recurring operating cost. With millions of edge interactions, even a small rise in bandwidth usage can translate into noticeably higher monthly bills [3]. What was once a capital expense has effectively become a continuous subscription. Federated learning (FL) offers a way out of this cycle. It keeps information where it is produced and sends only the model updates to a coordinating server [4]. Doing so limits the movement of sensitive data and can reduce network traffic [5]. The trade-off between bias and cost is a coordinating effort. Every training round requires clients to synchronize and thus adds time and overhead. Earlier works have focused on privacy and algorithmic efficiency but cost has received less attention [6]. The economic efficiency of FL under commercial cloud pricing remains unexplored. FL can also be viewed as a social computing process. Edge devices behave like an individual participant, contributing its computation to the server and thus tries to attain a shared goal. The collective output of these clients determines the overall performance. As network grows the workload gets distributed across nodes depicting how collaboration functions in social systems. In this work, we measure this collaboration scheme translates into tangible economic outcomes. Using the California Housing dataset [7], we compare centralized and federated configuration under identical conditions. A pricing-based model is designed which estimates compute, transfer and storage costs using actual cloud billing rates. The analysis finds out the trade-offs between accuracy, cost, and runtime when computation shifts from centralized servers to network edge. The findings show that federated learning can provide competitive accuracy while cutting costs to a large magnitude. More importantly, the model illustrates how FL can operate as a practical, privacy-preserving, and social computing tool for sustainable cloud-based AI.

2. RELATED WORK

Federated Learning (FL) began to train models without transferring data into a single location [4]. The FedAvg approach which is mainly practiced in Federated Learning repeatedly combines local updates from many devices. It can produce accuracy close to that of centralized model. Privacy is automatically preserved in this approach. The framework has been refined several times to handle uneven and complex data distribution. Some researchers use reinforcement style approaches to selectively decide which clients should exactly participate in each round. Other works have worked

to design personalized aggregation rules that adjust to local characteristics [8, 9]. These refinements make convergence faster and help in maintaining stability even if the client data varies widely. Another set of work concentrates on security and privacy. It works on protecting the shared gradients from attacks which could reveal sensitive information. Differential privacy [10] is a popular method where random noise is injected such that individual records cannot be traced. Secure aggregation and homomorphic encryption are other two techniques which prevents the server from inspecting local contributions [11, 12]. These protections make FL safer but also heavier. They increase both communication and computation load. As a result, assessing the economic side of FL has become the logical next step for practical deployments.

2.1 Cost and Resource Efficiency in Federated Learning

Cloud economics now play a growing role in how federated learning systems are designed. One of the earliest studies [5] proposed a cost model centred on bandwidth use. Later work showed that choices such as how many clients participate, the size of each update, and how often rounds occur can all influence total spending [13]. Reducing communication has therefore become a major theme in FL research. Gradient compression and scheduled updates help lower payload size and make large-scale deployments more affordable [3]. Other studies have examined how pricing strategies affect performance. Spot Fed, for example, demonstrated that using cloud spot instances can reduce cost but may affect execution reliability [14]. A recent review [6] observed that, despite these advances, the financial aspects of federated learning are still rarely tested against real cloud prices. While most works focused on federated algorithms and privacy issues actual cost of running federated systems have hardly been touched. This creates a gap in the existing literature. The real economic feasibility when cloud providers bill for storage, bandwidth and compute time has never been tested in practice.

2.2 FL as a Social Computing Paradigm

Federated learning has wide application in several domains especially in areas where security and privacy issues with raw data plays a prominent role. Healthcare systems train models across hospitals without mobilizing patient records. Similarly, mobile applications use it to refine text predictions and personalization. It also plays key role in financial fraud detection and emerging smart transportation networks [2, 15–17]. In all the above domains each device acts as a contributor. It sends learned updates to a shared model. Through this collective process, the overall system improves and the participants continue to learn together. This technique gives a flavour of social computing, in which large groups coordinate to solve problems using distributed information. Within FL, both computation and bandwidth allocation emerge from the collective behaviour of clients. When participation is broad, individual devices shoulder more of the workload, easing the demand on the central server and cutting storage requirements. Our study builds on this view by showing, through real measurements, how collaborative training can redistribute costs from the cloud to the network edge. Such cooperation reflects a socially coordinated approach to managing computational resources in cloud–edge ecosystems.

2.3 Summary and Research Motivation

While strong progress has been made in algorithmic optimization and privacy techniques, quantitative evidence on whether FL offers real cost benefits in deployed cloud conditions remains limited. Only a limited body of work has evaluated federated learning using empirical execution traces instead of theoretical communication estimates, or incorporated actual cloud billing models tied to commercial pricing. Even fewer studies include a direct performance comparison with a centralized baseline under identical data and experimental settings. Motivated by these gaps, the present study performs a trace-driven economic evaluation of federated learning. We apply real communication and runtime logs to a transparent cloud-pricing model to quantify cost behaviour accurately, and we assess whether collaborative participation among edge clients enables federated learning to operate as a cost-efficient social computing mechanism for cloud resource allocation.

3. METHODOLOGY

3.1 Experimental Setup

The empirical study evaluates whether federated learning (FL) can reduce cloud expenditure compared to centralized training while retaining model accuracy. We adopt the California Housing dataset [7] consisting of 20,640 records with eight predictive features. The model used in both configurations is a feed-forward regression network implemented in PyTorch®, trained for 10 communication rounds (FL) or 10 epochs (centralized) for comparability. Two client devices simulate distributed data ownership by partitioning the dataset evenly (IID setting). FL is coordinated using a synchronous FedAvg aggregation scheme. All experiments are executed in a controlled environment using the same cloud-hosted runtime and hardware configuration as shown in TABLE-1.

Table 1: Experimental configuration and hyper-parameter settings used for all training runs.

Component	**Specification / Value**	**Notes**
Host Machine	Intel Core i5 (10th Gen), 8 GB RAM	Used for all experiments
Operating System	Windows 10	Local controlled environment
ML Framework	PyTorch 2.1.0	Model implementation
Federated Learning Framework	Flower (v1.x)	Synchronous FedAvg
Dataset Partitioning	IID split across 2 clients	Equal data distribution
Communication Rounds	10	Update frequency
Training per client per round	1 epoch	Matched compute load
Logging Pipeline	Custom runtime + transfer logging	Exported for cost analysis
Cloud Pricing Model	Applied to collected logs	Based on commercial billing

Prediction performance is measured using Mean Squared Error (MSE) and Coefficient of Determination (R^2), computed on a hold-out test set identical for both training approaches. For the federated configuration, two simulated clients each train locally for ten epochs and periodically transmit model updates to a coordinating server that performs FedAvg aggregation. Only the parameter deltas-not raw data-are communicated. This design enables a fair comparison with the centralized baseline

described next. The detailed system architecture of the implemented federated learning framework, including client–server communication and update aggregation, is depicted in FIGURE 2. The complete workflow of the proposed methodology-covering dataset preparation, centralized and federated training, execution logging, and cost evaluation-is illustrated in FIGURE 1.

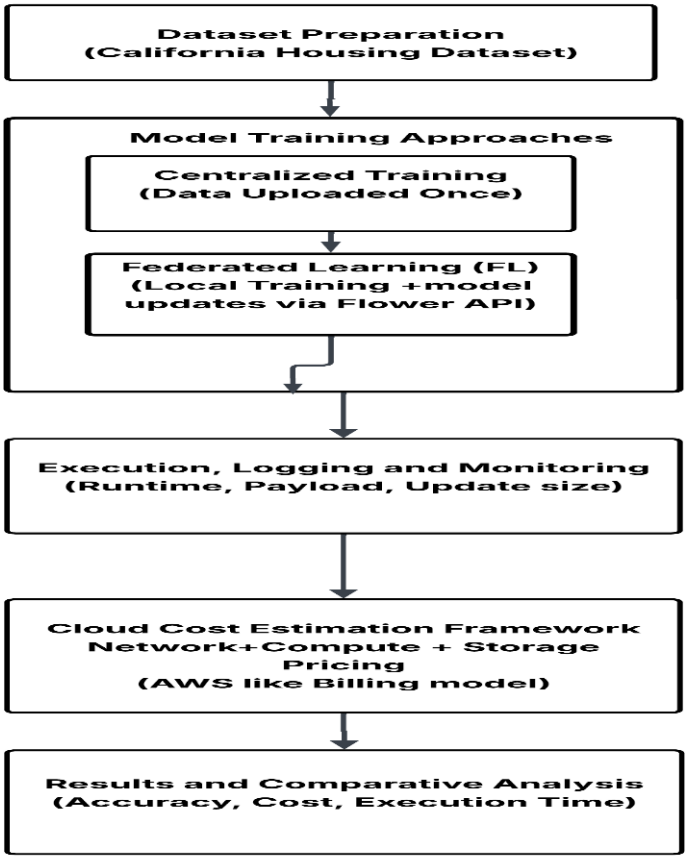

Figure 1: Overall workflow of the proposed methodology showing dataset preparation, centralized and federated training, execution logging, and cost evaluation under cloud billing. (Source: -Author)

3.1.1 Detailed experimental workflow

To complement the high-level methodology already presented, this subsection describes the precise procedural steps followed during execution and logging.

Step 1: Dataset handling and partitioning

The California Housing dataset was pre-processed and divided into two equal IID subsets. Each subset was assigned to a simulated client. Raw data was in the local client environment at all stages.

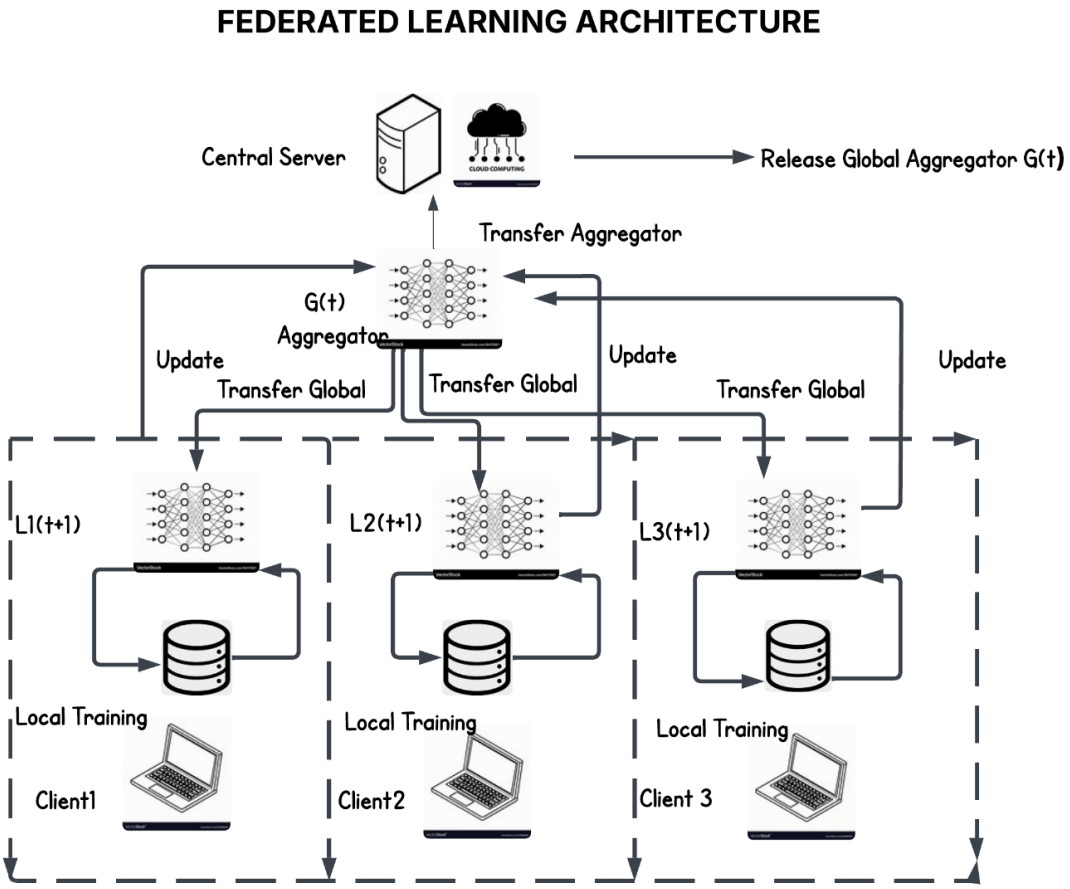

Figure 2: Federated learning workflow illustrating distributed client-side training and periodic aggregation on a central server. Dashed lines represent communication overhead that contributes to cloud network charges. (Source: - Author)

Step 2: Local training and update generation

Each communication round involved one epoch of local training per client. After training, clients generated serialized model-update vectors. The byte-size of each update was logged for cost estimation.

Step 3: Server-side aggregation and runtime logging

The coordinating server performed synchronous FedAvg aggregation. For every round, the important parameters were recorded like local training runtime per client, aggregation time on the server, size of incoming update payloads, temporary model checkpoint size.

Step 4: Cost-mapping using commercial billing rates

All recorded events such as data transfer, runtime, and temporary storage were mapped to realistic cloud pricing parameters. This trace-driven approach ensures that the economic comparison reflects actual usage behaviour rather than theoretical assumptions.

Step 5: Performance and cost comparison

Both configurations were evaluated on identical test data, using MSE and R^2 as accuracy metrics. The measured runtime and logged transfer volumes were then converted to monetary values to enable direct comparison under identical conditions.

This detailed workflow supplements the methodology without restating the diagram already provided as in FIGURE 1.

3.2 Centralized baseline

In the baseline setup, the full dataset is uploaded once to the cloud, where all computations run in a single environment. This setup depicts how production machine learning systems operate and thereby creates a benchmark for runtime and cost comparison. The centralized workflow is quite simpler in nature which involves transfer of data to the server and then training it as per the domain requirement. So, we see a single client server interaction phase in this process. Though this results in low communication overhead, but results in heavy bandwidth expenses. Frequent retraining with the arrival of new data results in more bandwidth expense. To have a consistent analysis experiments were carried out locally under controlled conditions. During each run every network payload and execution time was recorded. These measurement records were then mapped to real cloud pricing to estimate billing costs, to provide a practical data driven approach rather than a theoretical one.

3.3 Cloud cost modelling and metrics

To estimate the total monetary cost, we created a simple billing model that mirrors how commercial cloud providers charge for resources. Each component data transfer, computation, and temporary storage is linked to its corresponding price category. The total cost can be expressed as

$$C_{total} = C_{data} + C_{compute} + C_{storage}$$

Here, C_{data} represents the charges for data transfer, measured in gigabytes. In federated settings, this cost scales with the number of communication rounds. $C_{compute}$ denotes the cost of computation, billed per second of runtime.$C_{storage}$ covers the temporary saving of models and intermediate outputs. Transfer sizes and execution durations were taken directly from experiment logs, ensuring realistic accounting.

3.4 Network cost

To estimate the communication cost, we considered both data upload and download charges under standard cloud-billing terms.

For the centralized setup, the cost depends on the one-time transfer of the full dataset:

$$C_{data}^{cent} = S_{dataset} \times P_{in}$$

where $S_{dataset}$ is the total dataset size in gigabytes and P_{in} is the price charged per gigabyte of ingress.

In the federated setting, data are exchanged repeatedly over several communication rounds. The cost therefore accumulates over all participating clients and rounds:

$$C_{data}^{FL} = \sum_{r=1}^{R} (U_r \times N_c \times P_{out})$$

Here, U_r represents the model-update payload (in gigabytes) transmitted during round r.

N_c denotes the number of active clients.

P_{out} is the egress price per gigabyte.

R is the total number of FL rounds.

3.5 Compute cost

The compute component reflects how long a training job runs and how the cloud provider bills each second of use.

It is estimated as

$$C_{compute} = T \times P_{cpu}$$

where T is the measured runtime of the experiment, expressed in seconds, and P_{cpu} is the corresponding unit charge per second of computation. By tying runtime directly to cost, the model makes it easier to see how centralized and federated executions differ.

3.6 Storage cost

Temporary storage adds its own share to the total cost. It accounts for the short-term retention of model checkpoints and intermediate results created during training and aggregation. The storage expense can be expressed as:

$$\mathrm{C_{storage}} = \mathrm{V} \times \mathrm{P_{storage}} \times \mathrm{H}$$

Here, V is the volume of data held temporarily (in gigabytes), $\mathrm{P_{storage}}$ is the cost per gigabyte-hour, and H represents how long the data remain stored. However, the above-mentioned parameters largely impact the cloud cost. The model relies on actual experimental results. Actual measurements like transfer sizes, payload volumes and runtimes were collected during every experiment. So, cost estimations are grounded in terms of real usage and reflects commercial billing patterns instead of focusing on theoretical implications.

The cost model used in this study intentionally focuses on the three dominant pricing components such as network transfer, compute time, and temporary storage. As commercial cloud platforms

such as AWS, Azure, and GCP primarily bill users consider these factors. More complex pricing constructs e.g., reservations, spot markets, or multi-region replication were not included, as they vary by provider and deployment configuration. Our goal was to capture the fundamental economic behaviour of centralized vs. federated learning under controlled and comparable conditions. Since the experiment records actual payload sizes, runtime, and checkpoint volumes, this trace-driven model provides accurate relative cost differences even when the formulas remain simple. This ensures interpretability while reflecting the real billing mechanics faced by cloud users. Thus, the cost estimations remain grounded in real usage traces and reflect commercial cloud billing behaviour.

Table 2: Cloud pricing parameters - storage, compute-per-second, and bandwidth rates used for cost estimation.

Cost Component	**Pricing Value**	**Basis/Notes**
Data Transfer (egress)	$0.09 per GB	Charged for model update transfers
Data Transfer (ingress)	$0.09 per GB	Assumed symmetric billing
Compute Cost	Derived per-second from $2.49/month VM	Based on billed execution time
Storage Cost	$0.50 per GB per month converted hourly	Applies only in centralized setup

4. RESULTS AND ANALYSIS

4.1 Cloud Cost Comparison

TABLE 3 summarizes the estimated monetary expenditure for centralized and federated learning (FL), based on logged transfer volumes and billed execution time converted through the cloud pricing parameters in TABLE 2. The centralized pipeline required a single dataset upload to the cloud, incurring a large upfront charge for ingress transfer. In contrast, FL exchanged only small model update payloads in each communication round, resulting in orders-of-magnitude lower network charges even after repeated aggregation.

Table 3: Empirical cost and performance comparison between centralized and federated approaches.

Method	**Data Transferred (MB)**	**Estimated Cost (USD)**	**Execution Time**
Centralized	33.2	0.128	0.069
Federated (FedAvg,2 clients, 10 rounds)	0.004	0.000026	27.0

FIGURE 3 and FIGURE 4, present cost and time separately for clarity. Centralized training is costly because the entire dataset is uploaded once, federated training is nearly cost- free because only small model updates are transmitted. Conversely, federated learning takes longer due to

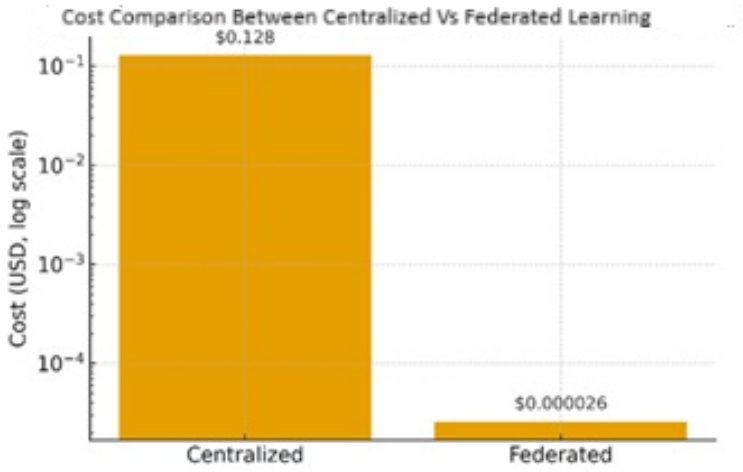

Figure 3: Comparative cloud cost between centralized and federated learning under identical billing conditions (log-scale view).

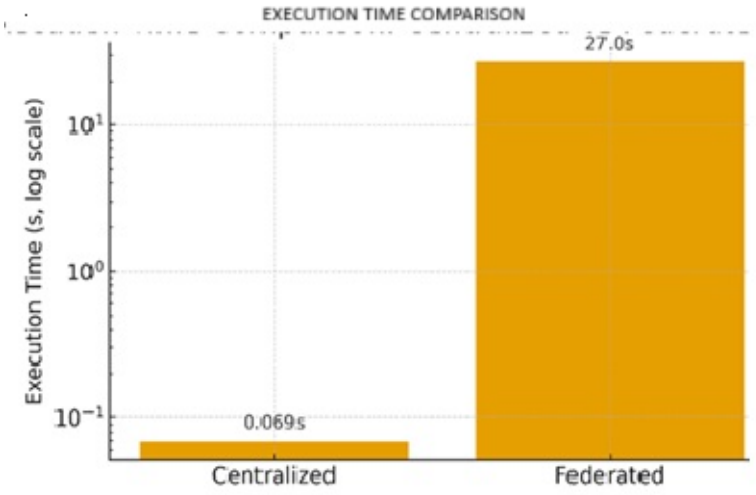

Figure 4: Execution-time comparison for centralized and federated training pipelines showing relative performance per communication round.

multi-round synchronization. These results confirm that cloud expenditure depends far more on communication volume than on compute duration in this setting. These results indicate that FL reduces cloud network charges by ~5,000×, even though wall-clock time is higher due to multiple communication rounds and model synchronization. The cost is dominated by bandwidth charges in the centralized case, whereas FL shifts cost toward collaborative computation among edge devices - a characteristic consistent with social computing principles. Overall, federated learning achieved a

~5,000× reduction in estimated transfer cost and maintained $R^2 \sim 0.59$, verifying its cost–accuracy balance.

4.2 Performance-Model Accuracy

To ensure economic advantages did not come at the expense of model fidelity, predictive quality was evaluated on the same test set for both configurations. FL slightly outperformed the centralized model (ΔR^2 + 0.011), confirming that keeping data decentralized did not harm - and may have helped - model generalization in this scenario which is depicted in TABLE 4.

Table 4: Summary of key metrics (accuracy, runtime, total cost) for the evaluated configurations.

Method	**MSE**	**R^2**
Centralized	0.556	0.576
Federated	0.542	0.587

4.3 Interpretation and Implications

These findings clearly demonstrate that economic efficiency in federated learning arises primarily from collaborative participation among clients. Instead of paying for full dataset transfers, cost scales with the size of transmitted model updates and the frequency of synchronization rounds. As a result, federated training enables continuous model refinement without repeated raw-data uploads, thereby shifting resource expenditure from centralized bandwidth to distributed client computation. Even though runtime is longer in FL (27.0 seconds vs. 0.069 seconds), the cost per execution is extremely small because the cloud charges scale primarily with payload volume, not time spent training locally. Viewed through a social-computing lens, each client lends its own computing power, easing the workload that would otherwise rest on the cloud. The central server lays off its heavy computation and just acts as a coordinator in the FL architecture. This methodology resembles a crowd system where many small contributions results in a larger outcome. Rather than massive centralized data transfer FL relies on smaller, distributed updates and thereby transforms expensive communication to practical collaboration. In real cloud deployments, this paradigm shift in machine learning technology creates a sustainable balance between computing performance and financial cost.

5. DISCUSSION

Existing federated learning deployments have predominantly focused on improving privacy and communication efficiency rather than quantifying explicit cloud-billing implications. The original Federated Averaging (FedAvg) framework by McMahan et al. emphasized communication-efficient deep network training using edge-held datasets [4], and subsequent improvements such as sparse updates, compression, and adaptive participation further reduced payload volume [18–21]. However, these works optimized communication primarily for latency, not economic cost. Several studies

have explored FL from the perspective of edge energy consumption and device-level resource budgets [22, 23]. While such models highlight that decentralization reduces cloud dependency, they do not translate runtime or data transfer into monetary terms. Similarly, surveys on FL in mobile edge computing environments describe system-level architectures but leave cost evaluation to future work [24, 25]. Only a small body of research has investigated cloud cost within distributed learning. Konečný et al. demonstrated that bandwidth egress substantially dominates operational cost in real cloud deployments [26], and Ding et al. highlighted communication-overhead pricing challenges under commercial billing [27]. This analysis, however, relied on assumption-driven models rather than execution-derived traces. Our study contributes a novel trace-driven cost analysis by converting measured synchronization payloads and execution times into realistic cloud pricing. The observed ~5,000× lower total data-transfer cost for federated learning directly supports findings that data movements, rather than CPU time, determine expenditure in commercial clouds [26, 27]. Unlike earlier simulation-only models, our results are derived from empirically logged communication under a fully operational FL pipeline. Accuracy comparisons show that FL reaches comparable or slightly better predictive performance compared to centralized learning ($\Delta R^2 + 0.011$). Our results align with earlier reports showing that preserving local data differences can improve a model's ability to generalize [28]. Looking at the problem from a social-computing perspective, federated learning spreads the infrastructure work across many cooperating clients instead of relying on a single cloud controller. Each client contributes part of the computation, which lightens the load on the central server and cuts down on large data transfers. FL set up behaves like a form of collective intelligence. In this scheme numerous distributed agents keep up the privacy concerns and efficiency of the system [29, 30]. However federated learning takes more time to execute as the parameters are exchanged repeatedly during many communication rounds. However, in future refinements could reduce this execution time. Several methods like adaptive scheduling [20], selective participation of clients [21], gradient compression [19], and asynchronous aggregation [18] are evolving methods to lower communication cost. More research is required to ensure a fair policy for participation of clients and to manage their energy use. It would also allow to maintain privacy as deployments grow to large scale.

5.1 Practical Considerations Beyond the Controlled Setting

Real deployments of federated leaning operate under condition which are different from that of controlled experiments. Clients differ in network speed, availability of resources, computational capability, and model sizes. Model sizes may range from small neural networks to very large architectures. These factors influence how FL behaves in practice but there is no change in the economic comparison. Cloud providers charge for data transferred, not the speed of transfer which in other words is termed as data ingress. Slow or unstable network links lengthen training time but do not increase bandwidth cost. Synchronization delays mainly add to server- side compute time, yet compute charges far lower than egress pricing in commercial cloud plans. Larger models increase the size of upload payloads in linear manner. However, modern FL systems commonly use techniques such as gradient compression [31], partial update schemes [32], or adapter-based modifications [33] to reduce the communication burden. Our controlled experiment was designed to isolate the monetary impact arising from data movement. The discussion above clarifies how the substantial savings in billing is attained by avoiding raw data upload. The above scheme extends to heterogeneous clients and different cloud providers. These considerations clarify how the ob-

served cost behaviour extends to heterogeneous networks, varying model sizes, and different cloud platforms.

6. CONCLUSION AND FUTURE SCOPE

This work examined federated learning under real cloud pricing scenarios. The data was kept local and only model updates were shared which sharply reduced network expenditure. Accuracy was preserved up to a large extent. The results suggest FL is a practical and economical solution for organizations managing sensitive and distributed data. Further research can refine client selection, compress communication, and adapt aggregation rounds to varying network conditions. Further testing is required with heterogeneous devices and multiple cloud providers. This will clear the scalability issues with FL scheme. To summarize FL not only safeguards privacy but enables cost-efficient and cooperative AI deployment. This is likely to be more popularized with time and is likely to shape the next phase of cloud-based machine learning.

7. ACKNOWLEDGEMENT

The authors sincerely thank the Department of Computer Science and Engineering at MCKV Institute of Engineering and Maulana Abul Kalam Azad University for their support and for making the computing resources available for this research.

References

[1] Saeed N, Ashour M, Mashaly M. Comprehensive Review of Federated Learning Challenges: A Data Preparation Viewpoint. J Big Data, 2025;12:153.

[2] Rieke N, Hancox J, Li W, Milletari F, Roth HR, et al. The Future of Digital Health With Federated Learning. NPJ Digital Medicine. 2020;3:119.

[3] Jiang Z, Xu Y, Xu H, Wang Z, Liu J, et al. Computation and Communication Efficient Federated Learning With Adaptive Model Pruning. IEEE Transactions on Mobile Computing. IEEE. 2024;23:2003–2021.

[4] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-Efficient Learning of Deep Networks From Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. PMLR. 2017;54:1273–1282.

[5] Konečný J, McMahan H B, Ramage D. Federated Optimization: Distributed Optimization Beyond the Datacenter. 2015. ArXiv preprint: https://arxiv.org/pdf/1511.03575

[6] Teixeira R, Almeida L, Antunes M, Gomes D, Aguiar RL. Efficient Training: Federated Learning Cost Analysis. Big Data Res. 2025;40:100510.

[7] Pace RK, Barry R. Sparse Spatial Autoregressions. Stat Probab Lett. 1997;33:291-297.

[8] Wang H, Kaplan Z, Niu D, Li B. Optimizing Federated Learning on Non-Iid Data With Reinforcement Learning. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE. 2020:1698–1707.

[9] Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A, et al. Federated Optimization in Heterogeneous Networks. Proceedings of Machine Learning and Systems. 2020;2:429–450.

[10] Dwork C Differential privacy. In Proceedings of the International Colloquium on Automata Languages and Programming. ICALP 2006. Lect Notes Comput Sci. Springer Nature. 2006;4052:1–12.

[11] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H B, et al. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017:1175–1191.

[12] Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, et al. Cryptonets: Applying Neural Networks to Encrypted Data With High Throughput and Accuracy. In Proceedings of the 33rd International Conference on Machine Learning. PMLR. 2016;48:201–210.

[13] Cho YJ, Wang J, Joshi G. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. 2020. arXiv preprint: https://arxiv.org/pdf/2010.01243.

[14] Shang R, Xu F, Bai Z, Chen L, Zhou Z, et al. spotDNN: Provisioning Spot Instances for Predictable Distributed DNN Training in the Cloud. In 2023 Proceedings of the IEEE/ACM 31st International Symposium on Quality of Service. IWQoS. 2023:1–10.

[15] Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, et al. Federated Learning for Mobile Keyboard Prediction. 2018. ArXiv preprint: https://arxiv.org/pdf/1811.03604.

[16] Liang Y, Guo Y, Gong Y, Luo C, Zhan J, et al. FLBench: A Benchmark Suite for Federated Learning. 2020. ArXiv preprint: https://arxiv.org/pdf/2008.07257

[17] Li Y, Tao X, Zhang X, Liu J, Xu J. Privacy-Preserved Federated Learning for Autonomous Driving. In IEEE Transactions on Intelligent Transportation Systems. 2022;23:8423–8434.

[18] Kairouz P, McMahan H B, Avent B, Bellet A, Bennis M, et al. Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning. 2021;14:1–210.

[19] Han S, Mao H, Dally WJ. Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization and Huffman Coding. 2015. ArXiv preprint: https://arxiv.org/pdf/1510.00149.

[20] Wang S, Tuor T, Salonidis T, Leung K K, Makaya C, et al. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. In IEEE Journal on Selected Areas in Communications. IEEE. 2019;37:1205–1221.

[21] Lai F, Zhu X, Madhyastha HV, Chowdhury M. Oort: Efficient Federated Learning via Guided Participant Selection. In Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation. OSDI. 2021:19–35.

[22] Nishio T, Yonetani R. Client Selection for Federated Learning With Heterogeneous Resources in Mobile Edge. In ICC 2019 IEEE International Conference on Communications (ICC). IEEE. 2019:1-7.

[23] Zaw CW, Pandey SR, Kim K, Hong CS. Energy-Aware Resource Management for Federated Learning in Multi-Access Edge Computing Systems. IEEE Access. 2021;9:34938–34950.

[24] Yang Z, Chen M, Wong KK, Poor HV, Cui S. Federated Learning for 6G: Applications, Challenges, and Opportunities. Engineering. 2022;8:33–41.

[25] Mao Y, You C, Zhang J, Huang K, Letaief K B. A Survey on Mobile Edge Computing: The Communication Perspective. IEEE Commun Surv Tutor. 2017;19: 2322–2358.

[26] Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, et al. Federated Learning: Strategies for Improving Communication Efficiency. 2016. ArXiv preprint: https://arxiv.org/pdf/1610.05492

[27] Ding N, Fang Z, Huang J. Optimal Contract Design for Efficient Federated Learning With Multi-Dimensional Private Information. IEEE J Sel Areas Commun. 2021;39:186–200.

[28] Liu L, Zhang J, Song SH, Letaief KB. Client-Edge-Cloud Hierarchical Federated Learning. In ICC 2020 - 2020 IEEE International Conference on Communications (ICC). IEEE. 2020:1–6.

[29] Gündüz D, Kerret P de, Sidiropoulos N D, Gesbert D, Murthy CR, et al. Machine Learning in the Air. IEEE J Sel Areas Commun. 2019;37:2184–2199.

[30] https://books.google.co.in/books/about/The_Wisdom_of_Crowds.html?id=hHUsHOHqVzEC

[31] Haddadpour F, Kamani MM, Mokhtari A, Mahdavi M. Federated Learning With Compression: Unified Analysis and Sharp Guarantees. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. PMLR. 2021;130:2350–2358.

[32] Wang H, Liu X, Niu J, Guo W, Tang S. Why Go Full? Elevating Federated Learning Through Partial Network Updates. 2024. arXiv preprint: https://arxiv.org/pdf/2410.11559

[33] Cai D, Wu Y, Wang S, Lin X F, Xu M. FedAdapter: Efficient Federated Learning for Modern NLP. 2022. ArXiv preprint: https://arxiv.org/pdf/2205.10162