How effective is federated learning at handling real-world data differences compared to centralized AI training?

Ahaan Gupta

Abstract

In the rapidly evolving era of artificial intelligence (AI), the increasing need for data privacy, low latency, and scalable model training made it an attractive alternative to conventional centralized training in the guise of Federated Learning (FL). This paper outlines how effective FL is in handling real-world data heterogeneity compared to centralized AI systems. Centralized AI models are based on data concentration at a central location, making them susceptible to privacy violations, latency limitations, and regulatory problems. FL, in contrast, trains models at the edge devices with data remaining localized, thus eliminating privacy and compliance problems. The distributed and frequently non-IID (non-independent and identically distributed) nature of data in FL, however, makes it challenging to train stability, converge, and be accurate. This paper comparatively evaluates these challenges through comparison of the two paradigms based on model accuracy, scalability, training stability, privacy, and deployment feasibility. A case study of FL in the healthcare sector illustrates the application of FL in real-world scenarios, detailing its advantages and trade-offs. The findings show that while centralized systems might be superior in laboratory settings, FL is a more practical and ethical solution for heterogeneous, decentralized, and privacy-concerned applications.

Literature Review

The distinction between centralized and federated learning structures has been the focus of many articles and studies in recent years. In centralized machine learning, data are gathered into a joint dataset, giving high accuracy and reliable training performance as long as data are IID (independently and identically distributed). Centralized systems enjoy benefits like continual optimization and faster convergence due to controlled data environments and infrastructure (GeeksforGeeks, n.d.; TechTarget, n.d.).

However, centralized training is typically plagued by scalability and privacy regulations, particularly in healthcare, where patient data exchange is restricted by legislations like GDPR and HIPAA. This limitation left a window with Federated Learning (FL), which was invented at Google in 2016, for training models on distributed clients without the sharing of raw data (Coin Monks, 2023). FL has been widely adopted in privacy-conscious settings, including mobile prediction and medical diagnosis (FeTS Initiative).

Studies such as Milvus (n.d.) and ScienceDirect (2021) note that FL is severely affected when case data are not IID. These limitations include convergence delay, model inconsistency, and fairness, especially when clients' data distributions are extremely heterogeneous. To counteract these limitations, researchers have proposed various countermeasures, such as clustering similar clients, personalized models, and robust aggregation techniques like FedAvg, FedProx, and Trimmed Mean.

From the perspective of system architecture, literature shows growing deployment of edge computing, which enhances the efficiency of FL through computing data closer to the generation site (GoodFirms, 2023). Although edge computing enables FL through reduction in latency and enhanced privacy, it raises resource complexity and synchronization.

The function of privacy-preserving mechanisms (PPMs) in FL is also determined. PPMs like k-anonymity, l-diversity, differential privacy, and secure multiparty computation are necessary to safeguard user identities and inhibit inference attacks (Li et al., 2021). These mechanisms generally create a trade-off between model accuracy and data confidentiality.

In summary, the literature indicates that FL is not without its boundaries—mainly in managing statistical heterogeneity and maintaining secure robustness—but is highly beneficial in real-world scenarios where decentralizing data and maintaining privacy are critical. As edge computing, adaptive learning algorithms, and secure aggregation protocols are increasingly being integrated, FL's usability and applicability remain on the rise in contemporary AI environments.

Introduction

The rise of artificial intelligence (AI) and machine learning (ML) has revolutionized industries across the globe. From diagnosing diseases in healthcare, to fraud detection in finance, to personalizing experiences in mobile applications, ML models have become essential tools. These advancements, however, come with a significant challenge: the need for vast amounts of high-quality data. Traditionally, data is centralized—collected from users or systems, stored on servers, and then used to train machine learning models. This method, while effective, introduces critical concerns such as data privacy violations, security risks, regulatory non-compliance (e.g., with GDPR or HIPAA), and inefficiencies in data transfer and computation. These challenges have led researchers and industries alike to explore alternative approaches that better align with the needs of today's decentralized, data-sensitive world.

One such approach is federated learning (FL)—a decentralized method of training machine learning models without the need to share raw data. Originally introduced by Google in 2016 to improve keyboard prediction on Android devices, federated learning enables individual devices or organizations to train a model locally using their own data. Instead of sending sensitive data to a central server, participants only share model updates or gradients. These updates are aggregated by a central server to form a global model, which is then redistributed for further local training. This cycle continues until the model reaches a desired level of performance. By keeping data localized, FL significantly enhances data privacy and compliance while enabling collaborative learning across geographically and organizationally distributed systems. The implications of this approach are profound, especially in domains where data is highly sensitive or legally restricted. In healthcare, for example, FL allows hospitals to collaboratively train diagnostic models without ever sharing patient data, preserving both privacy and compliance. In finance, it enables banks to improve fraud detection while keeping customer information confidential. In mobile and IoT applications, FL allows for real-time personalization—such as predictive text or smart device behavior—without uploading personal usage data to the cloud. Despite these benefits, federated learning introduces a new set of challenges, many of which stem from the heterogeneous nature of real-world data.

In theory, centralized learning assumes that all training data is IID—independent and identically distributed. But in practice, real-world data is often non-IID. For example, user behavior on smartphones varies drastically across individuals; medical records differ across regions or hospitals; sensor data in manufacturing depends on machinery types and usage conditions. This data heterogeneity means that each device or system participating in federated learning may have significantly different distributions of data, leading to biased or inconsistent model updates. These differences make it more difficult for the global model to generalize effectively, potentially reducing its accuracy and fairness across clients. Moreover, federated systems face technical and logistical challenges beyond data heterogeneity. These include communication overhead due to frequent model update transfers, disparities in device computational power (also known as system heterogeneity), and security concerns such as backdoor attacks or model poisoning. Even though FL is designed to protect data privacy, the updates sent to the central server may still leak information if intercepted or reverse-engineered. Techniques like differential privacy, secure aggregation, and robust aggregation

algorithms are being researched to address these vulnerabilities. Additionally, performance metrics for FL systems must take into account not only accuracy, but also fairness, efficiency, and resilience to faulty or malicious clients. Given the increasing adoption of federated learning in high-stakes, real-world scenarios, understanding its ability to handle real-world data differences becomes crucial. While the FL approach offers numerous benefits over centralized systems—particularly in privacy, scalability, and decentralization—its success is not guaranteed across all types of datasets or environments. The presence of non-IID data can hinder model convergence, degrade accuracy, and even compromise user trust if the system performs poorly on specific clients.

This essay investigates the research question:
"How effective is federated learning at handling real-world data differences compared to centralized AI training?"

By exploring the concept of federated learning, examining its uses across various industries, analyzing model performance under non-IID conditions, and comparing it with centralized training approaches, this essay aims to provide a balanced and detailed evaluation of federated vs conventional training. Due to the rise of AI for everyday activities it is important to look at it from, not only from a technical perspective but also from an ethical and societal perspective—especially since it is not just homework and text generation that AI is being used for. Ultimately, this research is significant because it explores a paradigm of AI development that attempts to balance performance efficiency with privacy. As federated learning continues to evolve, the question of its efficiency and effectiveness under real-world non-IID conditions will help to understand whether it can truly be as scalable and as effective an alternative to centralized AI systems.

Federated learning, based on its characteristics and functioning has aspects that make it better than the standard centralized system and have certain drawbacks compared to it as well. Majorly, the federated learning model is used to preserve privacy and information security in situations where storage resources or computational power is limited. This is due to the fact that it inhibits the requirements to have a centralized server to do the processing and compiling of the data. This allows for smaller and more distributed systems to contribute to a global model such as large and distributed hotel or even bank chains. Not only this but it allows to create networks for less powerful devices like smartphones or even industrial level or larger scale sensors that record regular data that can be used to train a model. This allows the system to have more personalization, diverse range of information and less training time and funding. Additionally, this system also enables cross industry collaboration such as a bank and a hotel that have overlapping customers but not the same kind of information about the customers, allowing them to make a more informed and comprehensive model. This method of cross industry and cross branch collaboration is known as cross-silo. This cross-silo system may be legally or practically inaccessible in a centralized system making federated learning all the more desirable. Moreover the fact that federated learning sends only model updates and not the data itself which allows it to not only maintain privacy but also be able to adhere to different privacy regulations like GDPR(EU law for protecting privacy and security for Europeans) and HIPAA(US law for protecting sensitive patient health information). Apart of the various benefits associated with computation requirements, storage requirements, bandwidth requirements it also makes training much faster since training is occurring across multiple clients in parallel as compared to the standard sequential processing method across a large dataset. This is one of the main reasons why the computational requirements is less since it is split across all devices. Federated training systems also follwo a few standard protocols associated with privacy protection. The standard is ofcourse FedAvg where only model updates are sent to the global model. Another is Secure aggregation where cryptographic algorithms or encryption is used to summarize the dates without showing individual contribution. These methods for example allow the federated learning system to outweigh the centralized system by allowing for safer and more private communication of information across nodes.

On the other hand, when it comes to the disadvantages of federated learning, the main problem arises due to heterogeneity. Different kinds of heterogeneity exist such as model heterogeneity, where each client essentially trains a different kind of model during the collaborative training, usually differing in terms of model size or architecture. This causes the global model to fit some nodes while losing out on the information of the other ones. Moving on, another type of heterogeneity exists which is communication heterogeneity, which occurs due to differing resources and speeds associated with network and bandwidth. It is possible that some of the clients have much faster network capabilities with higher upload and download speeds compared to other clients or systems. This makes the whole process less efficient and creates variation in terms of the rate of model updates being received from each system. Moving on, device heterogeneity refers to the difference and variation in hardware resources across different systems or clients. Some systems may have stronger and faster graphic units or central processing units (CPU) that may allow them to train the model at a faster rate and to be able to feed it more data compared to other slower or weaker systems. Finally, statistical heterogeneity, which refers to the non-uniform data distribution across the different clients, creates a challenge for one global model to be able to generalize well across all the information. This kind of heterogeneity is a result of non-IID data. Traditionally, federated learning expects data that is independent and identically distributed; however, that is not the reality, which essentially causes an unpredictable distribution of data. This heterogeneity creates slow convergence, reduced model accuracy, and increased communication overhead. Not only this, but the fact that the training is decentralized makes it vulnerable to data poisoning and inference attacks, since the data used for training is not inspected or standardized across all of the systems. Malicious nodes can introduce manipulated data or gradients that corrupt the shared model during aggregation. In addition to security vulnerabilities, federated systems often face fairness issues—certain clients may have significantly larger, higher-quality datasets or more computing power, allowing their updates to dominate training and skew the global model. Moreover, with nodes joining or leaving dynamically, known as participant instability, the consistency and reliability of training deteriorate further. Compounding this is the lack of standardization in data pre-processing across clients, which amplifies inconsistency in model performance and hinders reproducibility. This also means that the model may not work the same for all clients due to the difference it data distribution.

Understanding Real-World Data Heterogeneity

1) **Statistical (non-IID distributions among clients)**

Non-IID (non-independent and identically distributed) data in federated learning creates issues that compromise model performance, slow down convergence, and cause fairness issues. In federated learning, clients or devices learn locally from their data and upload updates to a central server. If data are non-IID—i.e., data distributions are very different across clients—global model performance generalizes poorly. For instance, one client learns cat images, and another learns dogs only, or sensor data from one area is very different from another. Such a mismatch violates the assumptions of vanilla machine learning, where models learn consistent patterns in inputs.

The most important effect is decreased model performance. Local models trained on unbalanced data send conflicting updates to the server. If you're training a word prediction model in which one user is writing technical reports and the other uses casual slang—the global model may overfit to the prevalent style or be unable to balance both. Moreover, non-IID data prevents convergence. In the vanilla federated averaging (FedAvg), the server is taking an average of the updates considering clients have the same data. If clients' data is different, their gradient updates are in opposing directions, necessitating more communication rounds for convergence. An example is a healthcare app collecting data from hospitals with different patient populations, which may train a reliable diagnostic model longer, increasing computational and communication cost.

Non-IID data also risks unfairness. Atypical data distributions of clients—e.g., the minority language in speech recognition—may exhibit poor personalized performance. This is because the global model is biased towards common patterns. Mitigate this by adding regularization methods to avoid local overfitting, clustering clients by data similarity, or personalized federated learning where all clients fine-tune the global model. For instance, a recommendation system clusters users by interaction history and trains cluster-specific models. Developers need to analyze data distribution among clients early and apply algorithms tailored to heterogeneity, e.g., adaptive optimization methods or weighted aggregation across data quality. Handling non-IID data is required to make federated learning reliable in real-world applications where data diversity is inevitable.

2) Systemic Differences

When we speak of data storage and retrieval very often you will hear those kinds of terms like hardware & cloud; but — they are different. Hardware refers to physical equipment such as hard drives, servers and computers. Similar to your local storage, it is on the device you are using. The cloud, of course, is storing data on the internet and are offsite servers hosted by such companies as Google or Amazon or Microsoft.

Hardware's storage, as one would guess from the name itself, are storage devices used for storing or porting or retrieving data and information and allow the users to directly interact with the computers. Desktop computers, laptop computers, mobile phones, servers, and data center hardware are some examples of hardware. Hardware may be costly to procure and maintain. It is space, power, cooling, and regular maintenance that are required in order to maintain it in tip-top working condition.

Hardware is costly to purchase and maintain. It occupies space, draws power, requires cooling, and will need to be serviced from time to time to maintain it in the condition that it should be. Hardware is not immortal and will eventually become outdated. What this means is that firms will have to replace their hardware constantly so that they can keep it current with the demands of present-day computing. Hardware is unchangeable and inflexible. Once installed, it is difficult and expensive to modify or upgrade.

Cloud storage, so called, is offsite storage whereby online data or information is stored on the internet through the use of cloud computing firms and contains files, documents, digital assets. Cloud examples are Amazon Web Services (AWS), Microsoft Azure, or Google Cloud. Cloud computing offers excellent scalability and flexibility. Users can merely add or remove computing resources depending on fluctuating needs without the need to buy extra hardware.
Cloud resources tend to be less expensive than equipment. Organizations only pay for what they use, unlike forking out for costly equipment. Cloud resources tend to be safer than equipment. Cloud providers invest a lot of money on security features to protect their customers' information.

3) Privacy Restrictions

Privacy Preserving Mechanisms PPMs are employed to protect user's sensitive and private information. Typically, we employ a sensitive attribute (SA) in the event we have user-specific private information that can be employed for research/statistical analysis, but must remain non-linkable to the concerned user. A quasi-identifier (QID) is a non-sensitive attribute (or set of attributes) that can be merged or combined with external/background information in an attempt to re-identify the individual to whom information relates. Finally, a key attribute is an explicit identifier (ID) of an individual, or alternatively, personally identifiable information (PII).

To ensure the privacy of the users, PPMs prefer to employ one or more data sanitizing operations such as generalization, suppression, perturbation, anatomization, permutation and slicing. Sanitization aims to protect sensitive information by removing or modifying data attributes. Apart from sanitization, PPMs can employ cryptography for ensuring the privacy of the data. These mechanisms employ protocols to facilitate distributed processing, sharing and retrieval of data with privacy guarantees.

One of the most widely used PPMs is k-anonymity which guarantees that in a set of k individuals, the identity of one among them cannot be disclosed from at least ????−1 individual belonging to the same set. The set of k individuals is referred to as an equivalence class. Also, the level of privacy achieved can be measured by the value of k, i.e., a higher value of k means a higher level of privacy. The l-diversity mechanism enforces k-anonymity and extends it by ensuring that every equivalence class is a set of entries such that there are at least l "well-represented" values for the sensitive attributes. However, l-diversity is marred by the drawback of adversarial knowledge assumptions. This mechanism assumes that in a case where the distribution of the attribute is known, the adversaries would acquire knowledge regarding a sensitive attribute, which is one of the drawbacks of this mechanism.

From l-diversity and slicing, Li et al. proposed a PPM for transactional data and structured data, referred to as l-diverse slicing. This mechanism weakens an adversary's ability to reveal the sensitive data of any individual with a probability greater than 1/l. For that, the attributes are partitioned into columns, column generalization is subsequently done by the algorithm and the tuples are partitioned into buckets. The highly correlated attributes are in the same column to preserve the correlation among those attributes, and the inter-relationship of uncorrelated attributes is lost. So, this mechanism prevents the linkage among different columns. The research paper proposed a mechanism for structured data that can be applied to several SAs. This mechanism is based on anatomization and slicing, keeping the k-anonymity and l-diversity constraints.

Comparison between Federated and Centralised learning

Centralized Learning

Strengths:
Access to comprehensive information - all information in one place:

- Centralized learning allows organizations to gather all training data in one location (like a server or data center). This setup helps train very accurate machine learning models because the system has access to a large and complete dataset. For example, a hospital chain can collect patient data from all their different branches to create a better trained and accurate model to help predict diseases.
- Compared to federated learning, centralized learning can often achieve higher accuracy faster because it trains on complete, unified data, while federated learning works with scattered data across devices and systems causing not only higher collection time but having to keep account of difference in time periods, network heterogeneity etc.

Easier to update models:

- In centralized systems, it's easier to update models, debug problems, and manage performance because everything is happening in one place. There’s no need to coordinate many devices or worry about different data formats across systems. The data is independently and identically distributes(IID).
- As compared to federated learning, it deals with non IID data since all the different systems hold different types of data in different formats causing a potential problem in collating the data together and leading to data heterogeneity.

More efficient:

- Since all processing is done on a central server or cloud platform, the training can be done on large computational powerhouses like a Nvidia A100 leading to faster and more efficient model training. This allows training to happen faster and more reliably than on devices with limited power (like smartphones or sensors).
- Moreover, the fact that all the different devices part of a federated system dont have the same computational power lead to discrepancies in the model training and some system may not be able to efficiently run the new updated global model as well.

Weaknesses:

Privacy and security risks:

- One of the biggest issues with centralized learning is that all the data must be moved to one place. This increases the chance of data leaks or hacking. Not only the transit of data but the storage in one facility also opens it up to attacks that can easily steal large amounts of data in one go. For example, storing all user health records in one server makes it a prime target for cyberattacks.
- Federated learning prevents this since only model updates are sent and not the data itself, preventing attackers to intercept data mid transit.

Legalities and bureaucracy

- In many places, especially in Europe under GDPR, collecting and storing user data in centralized servers is restricted. Healthcare and banking sectors are particularly affected because of strict privacy laws. Not only does this open the door to compliance costs, but this opens the need to have more investment to create safer and more rigid systems that prevent data leakage.

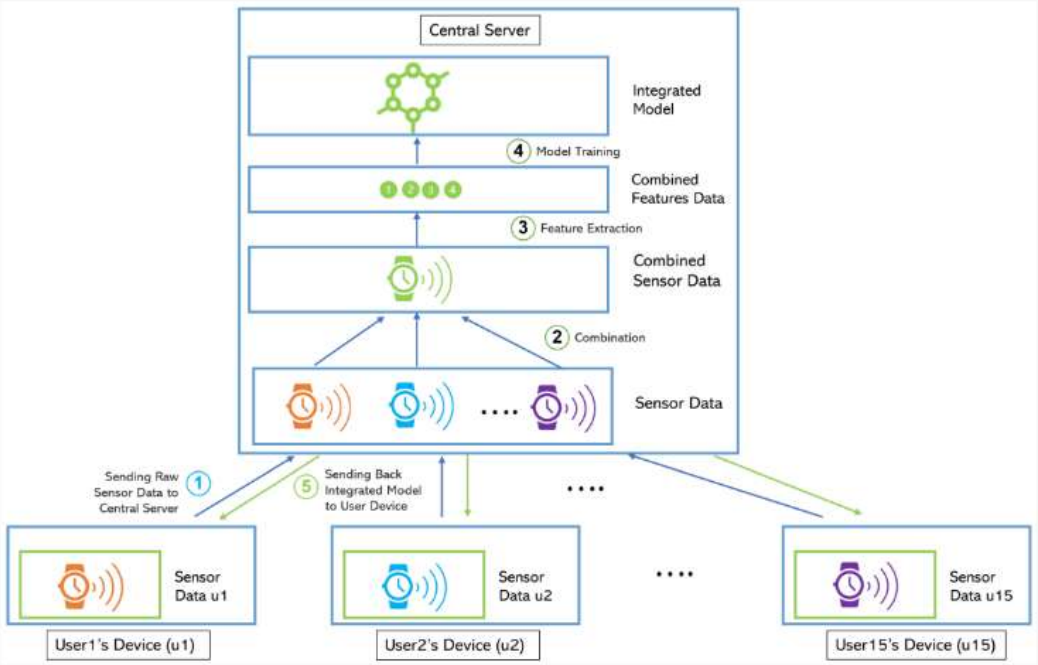
Dependant on a single server/system

- Centralized systems can break down if the central server crashes or is attacked. This affects all users and stops the learning process. A power outage or server bug could shut down the entire system. Any cases of data loss, data corruption or breaches would lead to the loss of all the data unless a backup is present. While this can be very easily prevented with redundant backups, but is still a risk since the whole range of data is at risk at the same time.
- Federated learning happens across different devices so it is much better in this case since even if one device fails there are still multiple other systems training the model.

High data transfer costs

- Moving large datasets from user devices to a central server takes time and bandwidth. In areas with weak internet or expensive data plans, this becomes a major problem. Training models with huge datasets (like videos or medical scans) becomes slow and costly. This has the drawbacks of federated learning since this involves delocalized systems sharing information to a central system. So all the different heterogenities such as system, data, network all play a role here and affect the efficiency of data transfer.

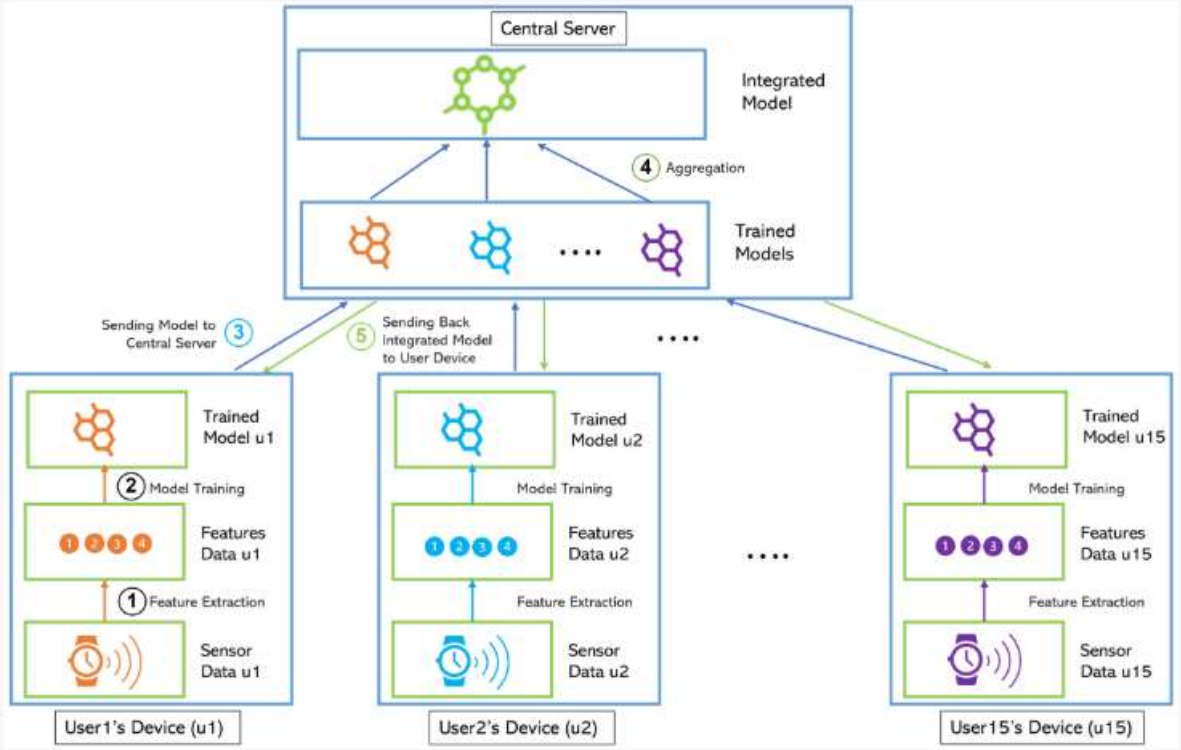
Figure 2. Centralized Learning Scheme.



As in the diagram above

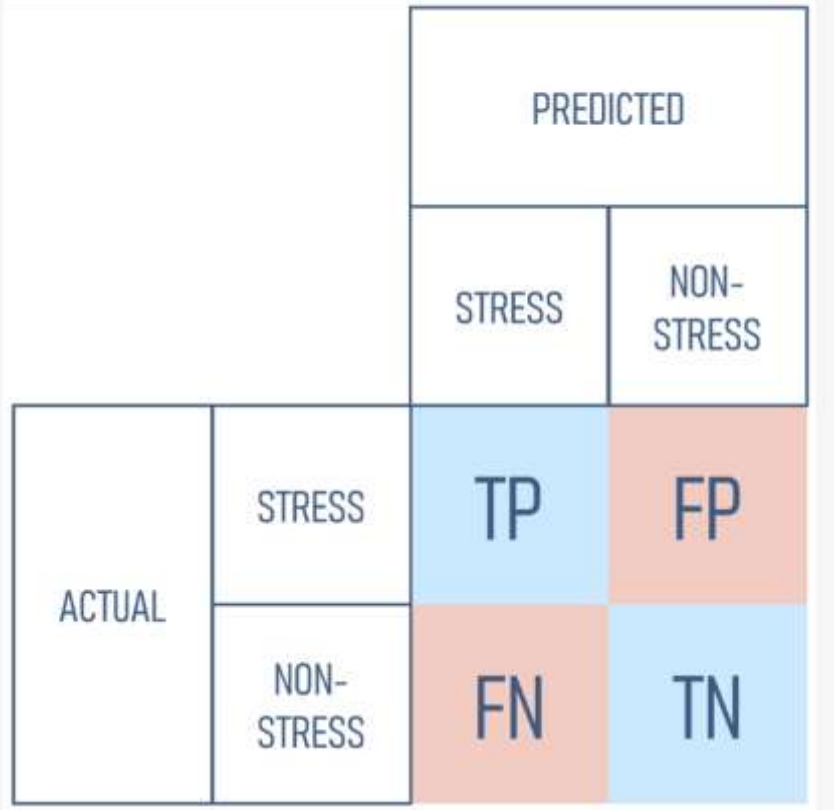
Each user's device (u1 to u15) collects sensor data and directly transmits the full raw dataset to the central server. This introduces significant privacy risks, since all personal data travels over the network and is stored centrally. The server aggregates all incoming data from users into one combined sensor dataset. This ensures high data volume and diversity but creates a single point of failure in terms of security. The server extracts features from the combined dataset. This could include patterns, trends, or variables useful for model training. A global integrated model is trained on the combined feature dataset using centralized compute power (e.g., GPUs). This step benefits from having a unified dataset, which helps in producing accurate and consistent results, assuming high-quality data. Once trained, the final integrated model is sent back to the individual user devices for use (e.g., predictions, monitoring, personalization). However, users do not contribute to model improvement after this unless they send more data again.

Figure 3. Federated Learning Scheme.



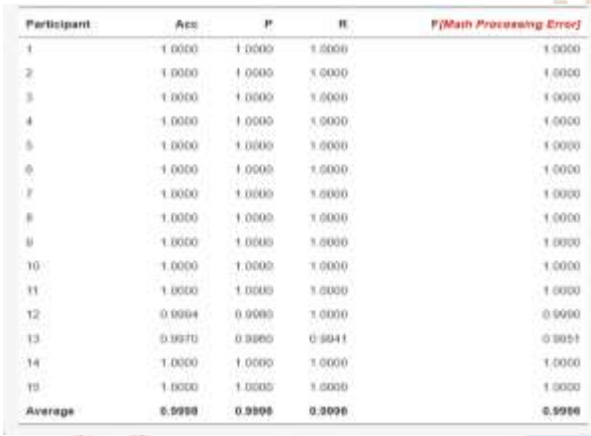
Each user's device collects sensor data and processes it locally to extract features (Step 1). This keeps raw data on the device. Based on those features, each device trains its own local version of the model (Step 2). These are the trained models u1, u2, u15, etc. The local models (just the updates/weights—not the data) are sent to the central server (Step 3). The server aggregates these local models using algorithms like FedAvg to form a single global model. The global model is then sent back to each device for further local training or use (Step 5)

Figure 4. Confusion Matrix. Blue square means the data are correctly predicted while red square means the data are incorrectly predicted.



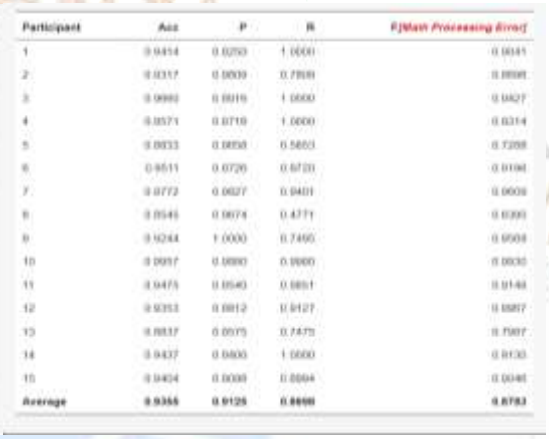
This matrix shows how well a classification model (stress vs. non-stress) is performing. It uses:

- TP (True Positive): Correctly predicted stress.
- TN (True Negative): Correctly predicted non-stress.
- FP (False Positive): Non-stress incorrectly predicted as stress.
- FN (False Negative): Stress incorrectly predicted as non-stress



Participant	Acc	P	R	F(Math Processing Error)
1	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000	1.0000
3	1.0000	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000	1.0000
6	1.0000	1.0000	1.0000	1.0000
7	1.0000	1.0000	1.0000	1.0000
8	1.0000	1.0000	1.0000	1.0000
9	1.0000	1.0000	1.0000	1.0000
10	1.0000	1.0000	1.0000	1.0000
11	1.0000	1.0000	1.0000	1.0000
12	0.9994	0.9988	1.0000	0.9994
13	0.9970	0.9960	0.9941	0.9951
14	1.0000	1.0000	1.0000	1.0000
15	1.0000	1.0000	1.0000	1.0000
Average	**0.9998**	**0.9996**	**0.9996**	**0.9996**

Table 1



Participant	Acc	P	R	F(Math Processing Error)
1	0.9414	0.8750	1.0000	0.9161
2	0.9317	0.8889	0.7869	0.8998
3	0.9992	0.9919	1.0000	0.9827
4	0.9571	0.9710	1.0000	0.9314
5	0.8833	0.8858	0.5853	0.7258
6	0.9511	0.9726	0.9733	0.9196
7	0.9772	0.9827	0.9401	0.9609
8	0.9546	0.9874	0.8771	0.9280
9	0.9244	1.0000	0.7466	0.9569
10	0.9917	0.9890	0.9890	0.9830
11	0.9473	0.9540	0.9851	0.9146
12	0.9353	0.9812	0.9127	0.9807
13	0.8837	0.9575	0.7473	0.7307
14	0.9437	0.9400	1.0000	0.9130
15	0.9404	0.9099	0.9894	0.9046
Average	**0.9355**	**0.9125**	**0.8898**	**0.8783**

Table 2

Participant	Acc	P	R	F(Math Processing Error)
1	0.9131	0.8675	0.8089	0.8372
2	0.7565	0.9872	0.1670	0.2857
3	0.9969	1.0000	0.9887	0.9943
4	0.7259	1.0000	0.0589	0.1113
5	0.8511	1.0000	0.4447	0.6156
6	0.8700	1.0000	0.5484	0.7083
7	0.8578	1.0000	0.5227	0.6866
8	0.7796	1.0000	0.1835	0.3101
9	0.7820	1.0000	0.2781	0.4352
10	0.9390	0.9950	0.8016	0.8879
11	0.9524	1.0000	0.8337	0.9093
12	0.9097	0.9917	0.7123	0.8291
13	0.7620	1.0000	0.2288	0.3724
14	0.8880	0.9967	0.6232	0.7669
15	0.8778	1.0000	0.6110	0.7585
Average	**0.8575**	**0.9892**	**0.5208**	**0.6339**

Table 3

Source: https://www.mdpi.com/2075-4426/12/10/1584

Table 1 Individual learning:

- Almost perfect scores across all users:
  - Accuracy ≈ 0.9998
  - Precision, Recall, F1-score ≈ 0.9996

Table 4 presents the evaluation of machine learning models trained individually on each participant's data. The performance here is exceptionally high—participants consistently achieve 100% across accuracy, precision, recall, and F1 score. The average accuracy is 0.9998, and other metrics match closely. These results imply that when the model is trained and evaluated on data from the same individual (with no data mixing or generalization needed), it performs almost perfectly. This method likely avoids heterogeneity and overfitting issues since each model is specialized for its own dataset.

Table 2 Centralized learning:

- Average Accuracy: 0.9355
- Precision: 0.9125
- Recall: 0.8698
- F1 Score: 0.8783
- High variability between participants (Participant 5 had 0.8833 vs. 10 with 0.9957).

Table 1 showcases results from centralized learning where all participant data is aggregated into a single server for model training. The average performance drops compared to individual learning: accuracy (0.9355), precision (0.9125), recall (0.8698), and F1 score (0.8783).Some participants (like 5, 8, and 13) show notably lower recall, suggesting the model is having trouble detecting positive stress out of all the actual positive in more personalized or unique data distributions where a single dataset comprises of a range of unique and diverse datasets.

Table 3 Federated Learning

- Average Accuracy: 0.8575
- Precision: 0.9892 (very high)
- Recall: 0.5208 (very low)
- F1 Score: 0.6339
- Very high precision but low recall suggests it misses many true stress cases (many false negatives).

The results in Table 3 for federated learning show another drop in performance compared to centralised learning. The average accuracy is 0.8575, precision is high at 0.9892, but recall is low at 0.5208, with the F1 score at 0.6339. These results suggest that the models are very conservative, with very few false positives, since 98.92% of the time, they predict a positive stress and are correct. However, since the recall is at 0.5208, it means that the model is missing many stress cases, not detecting stress even when it's there.

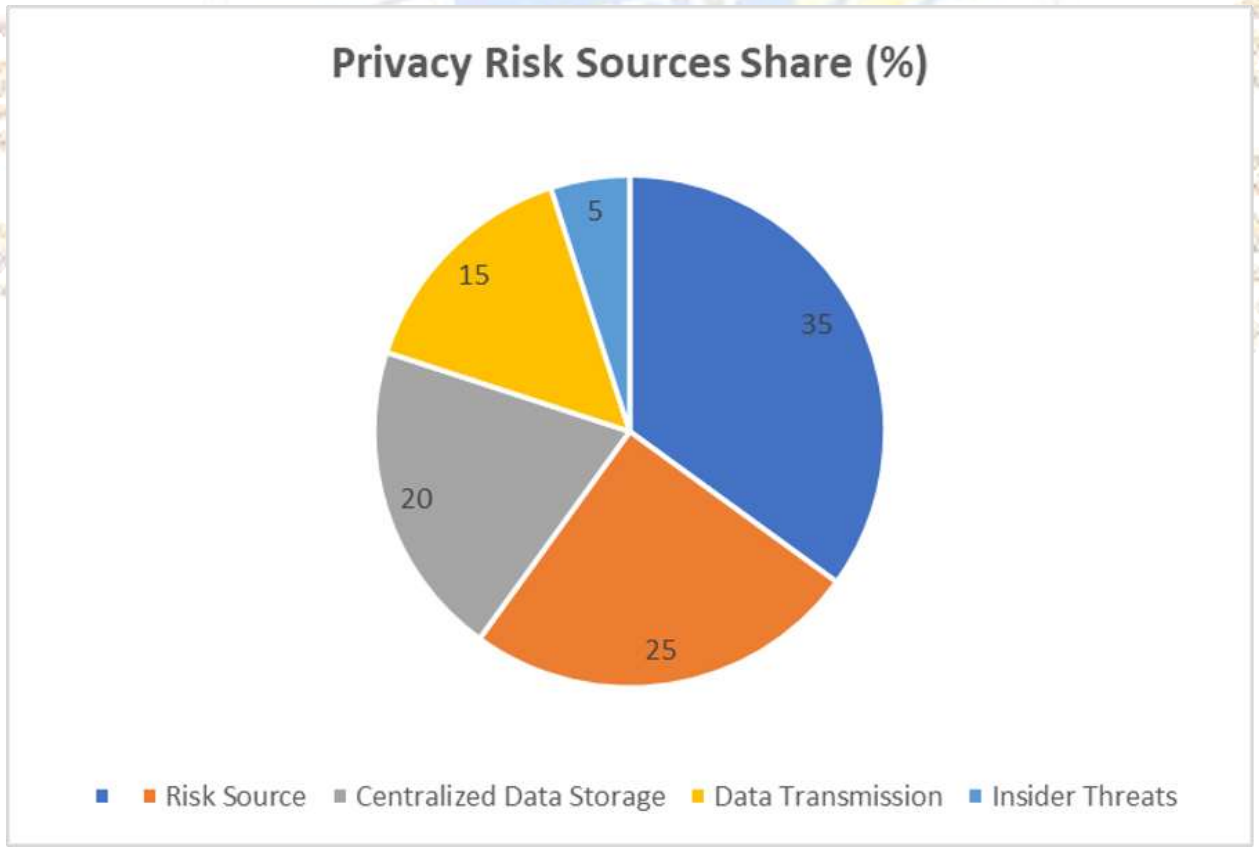
In other words, the model often fails to detect stress when it is actually present. This performance drop is likely due to issues inherent in federated learning, such as non-IID data (participants' data distributions are very different), device, communication, network and other heterogeneity. Additionally, when the central system attempts to consolidate all the various updates from devices, it can be challenging to create a model that works effectively for everyone.

Challenges Faced by Centralized and Federated Learning

First, communication overhead is a serious bottleneck. Federated learning involves frequent exchanges of model updates between devices and a central server, e.g., training a deep neural network (e.g., ResNet-50) in thousands of devices producing terabytes of data traffic, stressing bandwidth, and being more costly. Devices with weak connections (e.g., smartphones in poorly covered areas) may drop out, slowing down training. Model compression or lowering update frequency assists, but at the expense of precision or convergence rate. Developers need to strike a balance between efficiency and model quality, often necessitating customized protocols optimized for their hardware constraints.

Second, heterogeneity in data makes model convergence difficult. Data distributions in federated environments differ substantially across devices. For example, a typing pattern may be gathered by an app on a keyboard, making data non-IID. This can result in the global model performing poorly on a single device. A health app learned from data from multiple demographics may fail to generalize. Techniques such as Federated Averaging (FedAvg) do not work well with biased data, and sophisticated techniques (e.g., adaptive optimization or personalized models) introduce added complexity. Developers need to test for robustness on different data splits and avoid bias through careful sampling or regularization.

Third, security and privacy risks exist even with data stored on-device. Model updates are potentially exposing sensitive information; e.g., image classification gradient updates are potentially exposing identifiable features using inversion attacks. Adversaries also send poisoned updates to influence the global model (e.g., spam filters are poisoned to allow through malicious content). While methods like differential privacy or secure aggregation (e.g., encrypting aggregated updates) mitigate such exposure, they come at a cost. Privacy degrades model accuracy with noise addition, and crypto protocols add computation latency. Developers need to add protections without performance loss, and this typically includes extensive adversarial testing and iterative defenses.

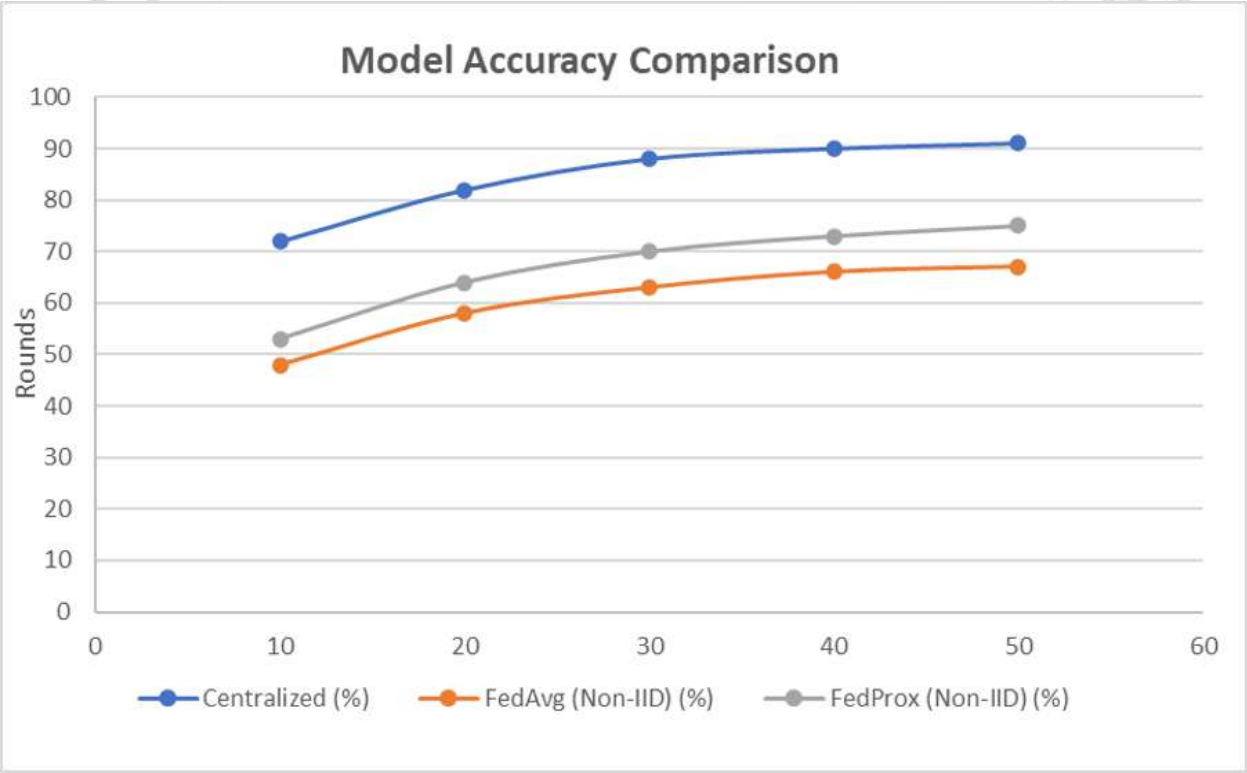


Source: Reported in breach studies by IBM Security's Cost of a Data Breach Report

Comparative Analysis: Federated Learning vs. Centralized Training

Centralized training of AI tends to perform well on diverse, large-scale data gathered in a single place. IID (independently and identically distributed) data is learned faster by centralized training and is usually more accurate as it is able to see the entire data distribution. However, when data are non-IID—i.e., unevenly distributed across sources (e.g., user devices, hospitals, banks)—centralized training needs all data gathered and preprocessed to reduce imbalance and bias. It is both expensive and infeasible because of privacy constraints. In addition, the central model may overfit majority data patterns and perform badly on underrepresented data types.

Federated Learning is designed particularly for training on non-IID data distributed over edge clients. In practice, however, model performance in FL is compromised due to client drift—local model updates of clients straying from the global optimum. Algorithms like FedAvg suffer when clients' distributions are extremely heterogeneous (e.g., language models on various regions or dialects). To counter this, recent work like FedProx, FedMA, and pFedMe has been introduced to counter statistical heterogeneity. Despite these improvements, centralized-level performance remains challenging in highly skewed environments, although personalized FL techniques are bridging the gap.



Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2017). *Communication-Efficient Learning of Deep Networks from Decentralized Data*. AISTATS.

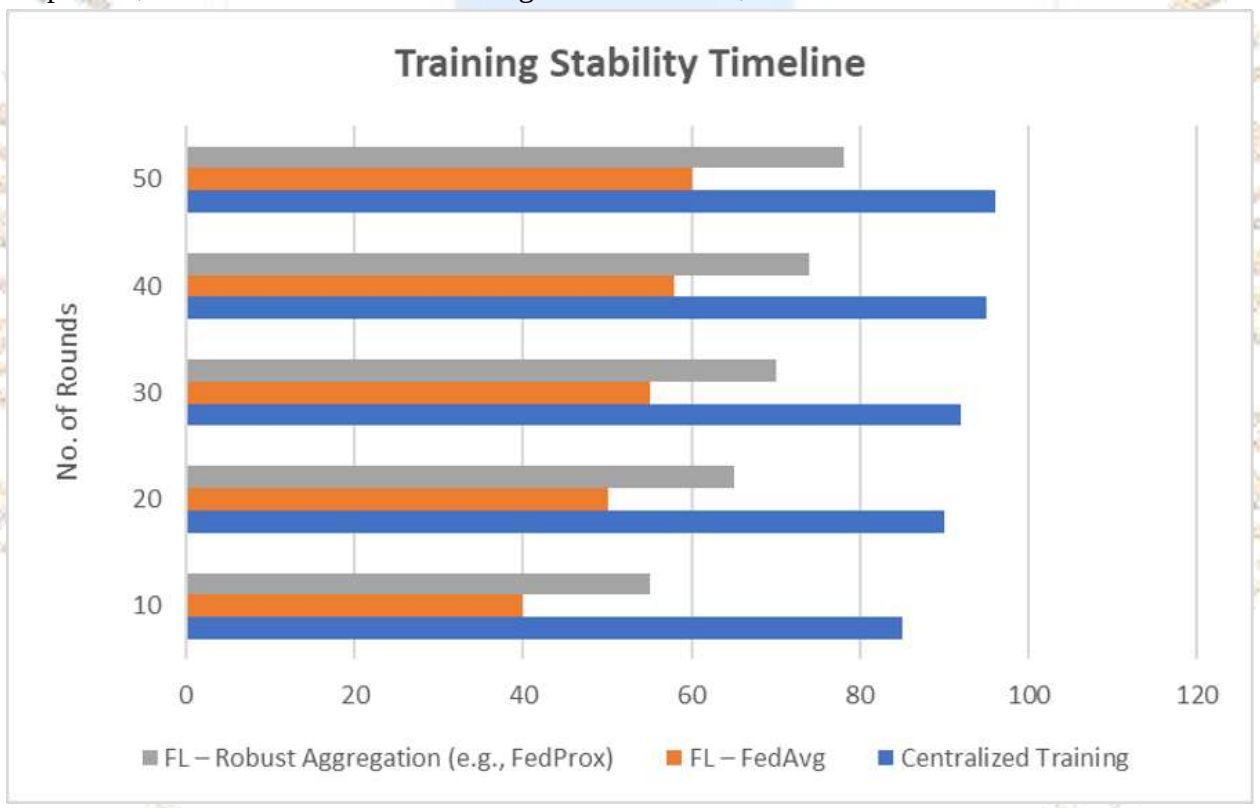
Stability in training is another essential aspect where centralized and federated solutions differ significantly. Centralized training enjoys the luxury of a well-ordered, homogeneous setup. Possessing all the data in a single setting ensures batches of training are well-balanced and gradient updates are uniform. Convergence in this context is likely to be uniform and smooth, particularly when models are trained with shared optimization methods like stochastic gradient descent (SGD) or Adam. Centralized systems are also simpler to debug and monitor because all training processes happen in a single, coherent framework.

Federated Learning introduces considerable complexity to training, however. Because each client trains locally on a separate dataset, and communication with the central server is asynchronous or intermittent, the training becomes more brittle. Clients can drop out when connection is lost, local updates can be skewed by small or unbalanced datasets, and hardware capability differences can result in slowdown of synchronization.

These cause higher training instability, especially for large federated networks. Unless the aggregation of client models is outlier- and poisoning-attack-resistant, the performance of the global model can also degrade. To mitigate these issues, robust aggregation techniques such as Krum, Trimmed Mean, and adaptive averaging have been introduced, although at the expense of introducing computational complexity into the system.

When scalability is considered, federated and centralized methods again show opposing strengths. Centralized methods are bounded by data transfer and storage limits. When the size of the dataset or the number of users grows, it becomes costlier and logistically intensive to retrieve and store data in the central server. Intensive computing resources need to be employed to handle the aggregated data, and centralized training becomes prohibitive for very large datasets. In addition, data transfer caps in privacy policies like GDPR and HIPAA can impede data transfer from its original location, adding another problem.

Federated Learning offers a scale-out solution by pre-localizing the data and performing computation on the client devices. This not only reduces the volume of large data transfer but also utilizes the idle compute capacity of client devices. In theory, this renders FL extremely scalable in terms of data volume and user interaction. Real-world scalability, however, depends upon the efficiency of client orchestration. Processing hundreds or thousands of client devices—each with varying availability, network, and computation resources—is not a trivial task. Communication bottlenecks and synchronization delays can throttle the process. But successful large-scale deployments, such as Google's application of FL in predictive text on Android phones, indicate that if one has the right infrastructure, FL can scale to millions of users.



Source: Li et al. (2020). *Federated Optimization in Heterogeneous Networks*.

Security and privacy are the pillars of FL's popularity. Centralized training involves privacy risks as raw data needs to be sent to and hosted on a central server. In spite of anonymization methods and in-transit encryption, centralized systems are susceptible to data breaches, insider threats, and regulation compliance. Additionally, the centralized approach becomes a valuable target for attackers intent on stealing sensitive information or tampering with training results.

Federated Learning addresses all these issues by design. Since no raw data is ever sent off the client device, the exposure risk is significantly reduced. FL is not completely secure by default, however. Model updates, even anonymized, can leak sensitive information via channels such as gradient inversion or membership inference. Malicious clients can attempt to poison the training process with poisoned updates. To defend against such attacks, advanced privacy-preserving techniques such as differential privacy, secure multi-party computation, and homomorphic encryption have been integrated with FL frameworks. While these techniques introduce additional computational overhead, they significantly enhance the privacy guarantees of FL systems, and those systems are well adapted to sensitive applications such as healthcare and finance.

Lastly, deployment cost and feasibility are the determining factors for deciding which paradigm is most feasible to deploy in the actual world. Centralized training demands a lot of initial expenditure on data infrastructure like data lakes, storage servers, and high-end GPUs. It also involves periodic fees for data management, cloud services, and compliance monitoring. Although this setup is feasible for large institutions that possess sophisticated data infrastructures, it is not as feasible for small players or in low-connectivity environments.

Federated Learning reduces the computational burden, potentially at the cost of saving central infrastructure costs. Leveraging client-side compute and avoiding the need for massive centralized storage, FL reduces data management costs. But there is a cost to this benefit. Effective deployment of FL depends on the use of orchestration software like TensorFlow Federated or Flower to handle client selection, fault tolerance, and synchronization. Additionally, device diversity and power constraints on edge devices can render deployment of FL less practical in certain environments. With these issues, though, the legal and ethical advantage of local storage of data generally outweighs the added complexity, particularly in environments where privacy law is stringent.

Case Study

Healthcare: Predicting Medical Conditions For Hospitals

Healthcare is perhaps the most promising but challenging area for AI because of stringent privacy laws like HIPAA (USA) and GDPR (Europe). Conventional centralized methods in healthcare are to collect information from various hospitals into a single database. Although this will be a high-precision model, it is mostly relegated to legal and moral issues. Hospitals might not be willing or might not be in a position to provide raw patient data because of privacy reasons, liability, and data governance regulations.

To avoid these constraints, Federated Learning is being employed by some medical AI applications. A case in point is the Federated Tumor Segmentation (FeTS) initiative, which brings together over 30 medical centers globally to cooperatively train models to identify brain tumors without sharing sensitive imaging information. With FL, hospitals trained local models from MRI scans and transferred encrypted model updates to a central server, which aggregated them to create a global model. The results were promising: the federated model performed similarly to a centralized model trained on aggregated information, showing that FL could provide good performance without leaking data privacy. Importantly, FL allowed each hospital to retain control of its data, thus ensuring compliance with patient confidentiality legislation.

But difficulties were also pointed out in the study. The hospitals participating employed varied MRI scanners, convention on labels, and patient populations, and hence there was extensive data heterogeneity. Institutions had much larger sets of training samples compared to others, and communication latency would sometimes bring the training process to a standstill. These logistical difficulties necessitated the application of sophisticated FL methods like FedProx to regularize training and personalization layers in order to accommodate the model to local data features. Conversely, conventional centralized training in healthcare

tends to provide slightly better accuracy in homogeneous conditions but is seldom possible at the same level because of legal restrictions and logistics problems.

Therefore, in deployment scenarios where patient information is sensitive and dispersed, FL has a clear practical feasibility and ethical advantage, although it involves more advanced management of heterogeneity.

With the coming age of AI, both centralized training and federated learning are a matter of the type of data, privacy requirements, support infrastructure, and application scenario. Centralized AI works best in stable environments with homogenous, heavy datasets that are not regulated by privacy laws. Since they can handle full datasets, they tend to produce better model accuracy, faster convergence, and simpler deployment—provided the data can be centralized without creating ethical or legal issues. Federated Learning, in contrast, offers a compelling solution.

Its privacy-preserving architecture naturally translates to real-world applications in healthcare, finance, and mobile computing, where user data must remain local. FL, however, continues to be plagued by challenges such as non-IID data, system instability, and communication overhead. Despite these problems, recent advances in personalized federated algorithms, resilient aggregation methods, and edge computing platforms are slowly bridging the performance gap with centralized models. Finally, although centralized learning is ideal in controlled environments with homogeneous data and infrastructure support, federated learning is increasingly an emerging privacy-sensitive, scalable paradigm that fits into the realities of the world's data environments. Its ability to confront the realities of the heterogeneities of real-world data improves day by day, and it emerges as an essential part of future AI systems in a privacy-conscious, decentralized world.

Conclusion

With the coming age of AI, both centralized training and federated learning are a matter of the type of data, privacy requirements, support infrastructure, and application scenario. Centralized AI works best in stable environments with homogenous, heavy datasets that are not regulated by privacy laws. Since they can handle full datasets, they tend to produce better model accuracy, faster convergence, and simpler deployment—provided the data can be centralized without creating ethical or legal issues. Federated Learning, in contrast, offers a compelling solution.

Its privacy-preserving architecture naturally translates to real-world applications in healthcare, finance, and mobile computing, where user data must remain local. FL, however, continues to be plagued by challenges such as non-IID data, system instability, and communication overhead. Despite these problems, recent advances in personalized federated algorithms, resilient aggregation methods, and edge computing platforms are slowly bridging the performance gap with centralized models. Finally, although centralized learning is ideal in controlled environments with homogeneous data and infrastructure support, federated learning is increasingly an emerging privacy-sensitive, scalable paradigm that fits into the realities of the world's data environments. Its ability to confront the realities of the heterogeneities of real-world data improves day by day, and it emerges as an essential part of future AI systems in a privacy-conscious, decentralized world.

Bibliography

DcentAI (November 25, 2024). Decentralized AI vs. Centralized AI: Key Differences and Advantages

Isabella Agdestein (February 27, 2025). Federated Learning: A Comprehensive Analysis of AI Training Without Data Sharing

Venkataraman Natarajan Iyer (January 01, 2024). A review on different techniques used to combat the non-iid and heterogeneous nature of data in FL

Nathan Sebastian (June 26, 2025). Edge Computing - Benefits, Applications, Challenges, and Opportunities

Stephen J. Bigelow (Dec 08, 2021). What is edge computing? Everything you need to know

Milvus. What is the impact of non-IID data in federated learning?

geeksforgeeks (September 04, 2024). Difference between Hardware and Cloud

Mariana Cunha, Ricardo Mendes, João P. Vilela (August, 2021). A survey of privacy-preserving mechanisms for heterogeneous data types

Milvus. What are the main challenges of federated learning?