REVIEW ARTICLE

Federated Learning for Artificial Intelligence in Embedded Systems

Kanthavel Radhakrishnan[1], **Dhaya Ramakrishnan**[1,*] **and R. Adline Freeda**[2]

[1] School of ECE, PNG University of Technology, Papua New Guinea
[2] KCG College of Technology, Chennai, India

Abstract

Federated Learning (FL) which eliminates the centralized data storage requirement by facilitating model training on diverse edge devices is now a promising paradigm for decentralized machine learning (ML). Applications involving privacy-preserving Artificial Intelligence (AI), including wearable technology, IoT networks, and smart healthcare appliances, can particularly benefit from this solution in embedded systems. By using on-device local data from devices such as sensors, embedded controllers, and smartphones, FL keeps confidential information local, minimizing the data transfer cost and privacy risks. Potentiality, challenges, and key applications of FL integration with embedded systems are addressed in this paper. Device-to-device efficient communication, model updating, and trade-offs between model accuracy and computational resource limitations are some of the issues addressed. Also addressed in the paper are model aggregation, federated optimization methods, and their usage in edge-based AI in real-life applications. Problems with security, system reliability, and heterogeneous data in federated environments are also discussed in the paper. The extensive use of FL in embedded systems is one of the important developments in edge AI solution designing that is more scalable, secure, and privacy-conscious.

Keywords: federated learning, embedded systems, artificial intelligence, edge computing, privacy-preserving internet of things, machine learning at the edge, data privacy, decentralized machine learning.

1 Introduction

The increasing application of AI in embedded systems has transformed the operation of such systems dramatically, enabling real-time decision-making and enhancing device autonomy. Embedded systems—such as Internet of Things (IoT) devices, wearables, and smart sensors—previously heavily utilized cloud computing for training and inference activities. This centralized method, however, poses significant issues, including subjecting data to privacy threats, vulnerability to security attacks, high costs of data transmission, and the unavailability of bandwidth in distributed environments. Such

Academic Editor:
Dharmalingam Muthusamy

Submitted: 21 March 2025
Accepted: 21 May 2025
Published: 27 June 2025

Vol. 2, **No.** 2, 2025.
10.62762/TETAI.2025.440076

***Corresponding author:**
✉ Dhaya Ramakrishnan
dhayavel2005@gmail.com

Citation
Radhakrishnan, K., Ramakrishnan, D., & Freeda, R. A. (2025). Federated Learning for Artificial Intelligence in Embedded Systems. *ICCK Transactions on Emerging Topics in Artificial Intelligence*, 2(2), 91–115.

© 2025 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

concerns are especially crucial in privacy-conscious domains like health, smart home, and individualized services where real-time sensitive data is being produced.

FL addresses such concerns by enabling decentralized ML. Rather than sending raw data to the cloud, FL offers the possibility of local training of models on devices and model updates relayed to a central server [12]. This distributed architecture significantly reduces the transmission of sensitive data, enhances privacy of data, and reduces the likelihood of breaches. FL also removes locality issues with data by training models at the data source. This is very helpful for embedded systems operating in environments with limited processing power, connectivity, and storage. In the domain of embedded systems, FL offers new possibilities for AI deployment in the network edge. These systems can now perform on-device learning without compromising the confidentiality of sensitive data, as required by applications such as healthcare monitoring, autonomous driving, smart homes, and industrial IoT. By enabling local decision-making and model updating, FL can significantly reduce latency, enhance real-time responsiveness, and lower the need for continual cloud communication, further boosting the overall efficacy of embedded systems [2].

Despite its clear value, FL integration into embedded systems brings a variety of unique challenges. One of the key issues is device heterogeneity: embedded devices are very heterogeneous in terms of computational capabilities, storage, and networking [25]. Ensuring that FL models can be trained effectively on such diverse platforms without sacrificing performance or accuracy is a major challenge. Moreover, model aggregation becomes harder when local updates must be aggregated from devices producing non-independent and identically distributed (non-IID) or noisy data [21]. Communication overhead is also a major problem. The frequent transfer of model updates among edge devices and central servers may lead to intensive bandwidth usage, especially for mass deployments. Compressing the model, pruning the model, and applying differential privacy have been proposed for minimizing communication overheads and ensuring data privacy for model aggregation. Security is yet another equally critical issue: since updates to models are shared across networks, FL systems are vulnerable to adversarial attacks and model poisoning attacks and demand the creation of robust security protocols for real-world FL deployment [39].

The aim of this paper is to investigate the uptake of FL in embedded systems through an examination of its benefits, limitation, and latest technological advancements. We introduce a comprehensive review of FL models and their accuracy under constraints of embedded settings. Some of the discussion areas include privacy-preserving techniques, training strategies for models, federated optimization algorithms, and real-world applications such as smart healthcare monitoring, self-driving cars, and smart industrial automation [13]. We also address current challenges such as handling non-IID data, fault tolerance, and scalability—and present recent advances to address these challenges. Finally, we introduce future research directions for enhancing FL-based embedded AI systems in terms of next-generation edge computing solutions through AI.

2 Foundations and Architecture of FL for Embedded Systems

Embedded AI and Edge AI are revolutionizing the way intelligent services are deployed, especially in resource-constrained environments such as IoT devices, sensors, wearables, and mobile platforms. FL a golden opportunity to jointly train models over distributed embedded devices without centralizing sensitive data, offering it as an ideal technique for next-generation edge systems [3].

2.1 Need for FL in Embedded and Edge Environments

FL in the context of embedded and edge deployments addresses some of the inherent issues of modern-day distributed systems. As the amount of edge devices increases exponentially, it is not economically feasible to push huge quantities of raw data to central servers with bandwidth concerns, latency concerns, security, and privacy issues [29]. FL offers a decentralized solution in the form of enabling devices to learn shared models in a cooperative manner without disclosing local data. This paradigm shift is crucial in the development of efficient, secure, and scalable edge smart systems for multiple edge applications.

- **Data Privacy:** performing processing of informed information (such as medical or security surveillance) right at the edge device avoiding to route it all back into a remote location [38].
- **Communication Efficiency:** FL reduces the amount of raw data that needs to be transferred

back and forth with servers, which is important for systems where bandwidth is critical [4].

- **Personalization:** user-specific data may be privately learned by the models on devices.
- **Scalability:** Millions of other interested devices can also help enhance learning without swamping the cloud servers.

2.2 Challenges of Implementing FL on Embedded and Edge Devices

FL has many benefits for embedded and edge AI systems, but it is difficult to efficiently deploy in these environments. Compared to typical cloud-based setups, embedded and edge devices suffer from harsh resource limitations, varying connectivity, and diverse hardware environments. To make FL models deployed over distributed edge networks effective, efficient, and robust, these challenges must be overcome [32].

- **Hardware Constraints:** The devices possess limited memory, compute power, and battery life.
- **Network Instability:** Synchronous model updates are plagued by unstable network connectivity from edge devices.
- **Hardware Heterogeneity:** Computers are much more heterogeneous with respect to hardware organization (ARM, RISC-V, special ASICs).
- **Security Risks:** FL edge exposes models to greater threats of model poisoning and adversarial attacks.

2.3 Recent Technological Trends

Recent developments in edge and embedded computing technologies have profoundly influenced the development of FL frameworks. Developments in low-power hardware, lean model design, and distributed training paradigms are rendering FL more viable and beneficial for real-world edge applications. Attaining awareness of these emerging trends is crucial towards understanding how FL is evolving to tackle the challenges and needs of embedded AI ecosystems [10].

- **TinyML Integration:** Ultra-lightweight ML model that is deployable in low-RAM and storage devices, enabling on-device learning via FL.
- **Low-Power AI Accelerators:** Hardware platforms like Google's Coral TPU and NVIDIA's Jetson Nano accelerate deep learning computations on embedded devices at high energy efficiency.
- **On-Device Training:** Techniques like quantization and pruning allow embedded devices to fine-tune and update FL models locally without servers.
- **Federated Reinforcement Learning (FRL):** Combines FL and reinforcement learning to enable cooperative embedded systems like drone swarms and autonomous vehicles.

Embedded and Edge AI settings are significantly advantaged by the decentralized, privacy-protecting, and communication-effective character of FL. With improved hardware and more advanced optimization methods being developed, FL will be a key enabler for extreme edge intelligence in networks [24].

3 Key Applications and Case Studies of FL in Embedded Systems

FL applied to embedded systems is a giant leap towards creating more privacy-focused, secure, and efficient AI-powered devices. With increasing technology advancements, FL is poised to transform sectors such as healthcare, smart cities, the automotive sector, and the Internet of Things (IoT) by offering an even more scalable, fault-tolerant, and secure way of delivering AI at scale. FL has garnered considerable attention in recent years for application in embedded systems as it can support decentralized ML with security and privacy. Embedded devices, particularly IoT devices, wearables, and smart sensors, typically operate in environments with scarce resources, and therefore typical cloud-based AI systems are not viable due to latency, bandwidth, and privacy concerns. FL surpasses these constraints by allowing training of models on the local devices directly and sending updates to models over the network instead of transmitting the data, keeping the data private and saving data transmission costs.

FL was initially proposed by McMahan et al. [28], in which they proposed an approach to train ML models on a large number of decentralized devices without invading data privacy. The main idea of FL is that the local data remains on the device, and only model weight updates are exchanged with a central server for aggregation to build a global model. The approach is especially handy for embedded systems applied in sensitive areas such as healthcare monitoring [17], autonomous vehicle systems [27], and home automation [30]. Even with the advantages,

its adoption in embedded systems is faced with several challenges. Its greatest challenge is the heterogeneity of the devices, where they vary greatly in computation power, storage, and network connectivity. Embedded devices such as smartphones, wearables, and IoT sensors have high heterogeneity, and hence it's challenging to deploy FL models effectively on different platforms. Another major issue is that data is non-IID (non-independent and identically distributed) across devices. In most real-world scenarios, local data significantly differs, making it difficult to aggregate local updates into a uniform global model [25]. This is an even more serious issue in large-scale FL networks consisting of thousands or even millions of devices [7].

In addition to device heterogeneity and data distribution problems, communication overhead is another major problem too. Communication costs associated with model updates transmission from edge devices to the server can be enormous, especially in networks with large device numbers. To address this, authors have proposed techniques such as model pruning, update compression, and sparsification for reducing the communication overhead [10]. Besides, deployed FL systems also need to address potential security risks. Model updates transmitted over networks are susceptible to attacks by attackers like model poisoning and backdoor attacks, which undermine the integrity and confidentiality of the learning process [4]. Although FL inherently preserves privacy through local data retention, model updates can still potentially leak sensitive information. To promote privacy and security, numerous advanced techniques have been incorporated. Differential Privacy (DP) introduces noise scaled to the model updates to prevent information leakage [37]. Secure Multi-Party Computation (SMPC) enables various parties to jointly compute model updates without exposing their private inputs [8]. Resilient aggregation techniques and adaptive techniques have also been developed to safeguard against model poisoning and backdoor attacks [18]. These methods guarantee that sensitive data are safe even in hostile settings, and this makes FL implementations in embedded systems more reliable and secure.

There has been progress in recent years to improve the privacy, scalability, and efficiency of FL systems intended for the embedded environment. The integration of TinyML, which allows the running of ML models on low-resource devices such as microcontrollers, has rendered FL possible even with computation and power resources available [36]. TinyML combined with FL is critical to supporting decentralized intelligence on IoT devices and wearables. Another recent development is Personalized FL, where it is possible to personalize local models for personal devices and yet still contribute to the global model to solve device heterogeneity as well as enhance model accuracy. Additionally, cross-silo FL with fewer trusted nodes (e.g., hospitals or firms) and cross-device FL at huge scale on personal devices have arisen as two different operational modes of FL. Energy-efficient FL techniques are also being designed to reduce power consumption due to local model training and communication, which is crucial for battery-powered devices. Additionally, Federated Transfer Learning (FTL) has been proposed to enable the contribution of small and highly disparate datasets of devices efficiently, leveraging transfer learning principles to leverage the performance of the global model. Application areas of FL in embedded systems are abundant and revolutionary. In medical surveillance, FL enables training customized AI models on health wearables such that real-time diagnostics and prediction are feasible without the need to send patient data to cloud servers. Within the autonomous vehicle context, FL allows vehicles to locally train their AI models with unique driving data in a private manner, thus allowing real-time decision-making and continuous learning based on the diverse conditions encountered by the fleet [26]. In clever homes, device like thermostats, lights, and security cameras utilize federated gaining knowledge of two parent human choices and automate responses at the same time as safeguarding personal household facts. FL is applied in business contexts for predictive protection, tool tracking, and anomaly detection, wherein nearby statistics is processed through deployed sensors and devices to enhance efficiency and avoid malfunctions, all even as safeguarding unprocessed operational information from imperative servers [27].

Integrating FL into embedded devices provides huge benefits in terms of privateness renovation, facts protection, performance, and scalability. However, to completely harness its capability, considerable challenges related to device heterogeneity, conversation overhead, non-IID data distributions, and security vulnerabilities need to be addressed. Federated optimization methods, privateness-improving technologies which include differential privateness and stable multiparty computation (SMPC), and power-green computing

strategies like TinyML are topics of ongoing take a look at with full-size promise for development. As federated gaining knowledge of evolves, it possesses the ability to convert healthcare, transportation, smart homes, and business packages by offering privateness-keeping, scalable, and efficient options to standard centralized device mastering strategies.

FL is a distributed ML technique that allows multiple age units (such as smartphones, IoT units or built-in system) to train a global ML model to stay located on devices [19]. The model allows the FL model updates (gradients) to be sent to a central server, rather than transferring raw data on a centralized server for training where they are gathered to improve the global model. This approach preserves data privacy, reduces communication costs and ensures that sensitive information remains on the device, making it especially useful for health care applications, finance and IoTs. Important concepts for FL include:

- **Local training:** Each unit trains its model on local data without the need to share with Central Server or other devices.
- **Model aggregation:** After the local model is trained, updates (not raw data) are sent to a central server for aggregation. The process usually includes average model parameters or gradients (eg when using fedavg-algorithm).
- **Decentralized learning:** The FL machine enables a decentralized approach to learning, which is the opposite with traditional centralized learning methods.
- **Privacy protection:** Since raw data is never exchanged, FL naturally is more privacy protection than traditional ML models that depend on central data storage [6].

3.1 Historical Development and Milestones

FL emerged from the need to train models on data that could not be centralized due to privacy concerns or logistical issues. The concept was first introduced by Google researchers in 2016, as a way to enable ML on mobile devices without sharing user data with centralized servers. The initial work by McMahan et al. [28] introduced the Federated Averaging (FedAvg) algorithm, which became a foundational method in FL. Key milestones in the development of FL include:

- **2016:** Introduction to Federated Learning by McMahan et al. [28] For mobile devices, such as central technology for model aggregation with FedAvg algorithm.
- **2017:** The release of Open-Source frames such as TensorFlow Federated, which helped make the FL algorithm more accessible and easier to use for researchers and developers.
- **2018:** For the IoT and health care applications that expanded FL, researchers demonstrated the ability of FL for personal health services while maintaining the patient's privacy.
- **2019:** Important research to increase privacy, such as model updates ahead and the use of differential privacy to prevent the leakage of sensitive information during aggregation.
- **2020:** Edge computing and integration of FL with 5G networks, fast and more efficient model enables training and distribution for IoT system [25].
- **2021:** Connecting research and model toxicity attacks in secure aggregation, makes FL stronger for use in the real-world applications [41].

3.2 Basic Architecture for FL

FL Architecture is usually composed of three main components and the Figure 1 elaborates the Basic FL Architecture.

- **Customer equipment (Edge Device):** These are devices (eg smartphones, IoT devices or built-in systems) that host local data and do local training. Each device calculates the model update based on the local dataset. These updates are usually gradients or weights that reflect local data training.
- **Central server (agriculture):** Central Server receives updates from client equipment. It collects these updates to improve the global model. The aggregation can be done by techniques such as FedAvg, where the server keeps the average of weight or gradients from customers and updates the global model. Server can also handle the orchestration of the training process, such as planning when each customer participates in training, manages communication between customers and controls the synchronization process.
- **Communication network:** Communication networks are used to transfer updates between clients and servers. In FL, data privacy is ensured

by not transferring raw data. Instead, local model parameters or gradients are sent to the server for aggregation. Communication is usually periodic, after sending updates to customers after training in a certain number of ages or completed local workouts [14].

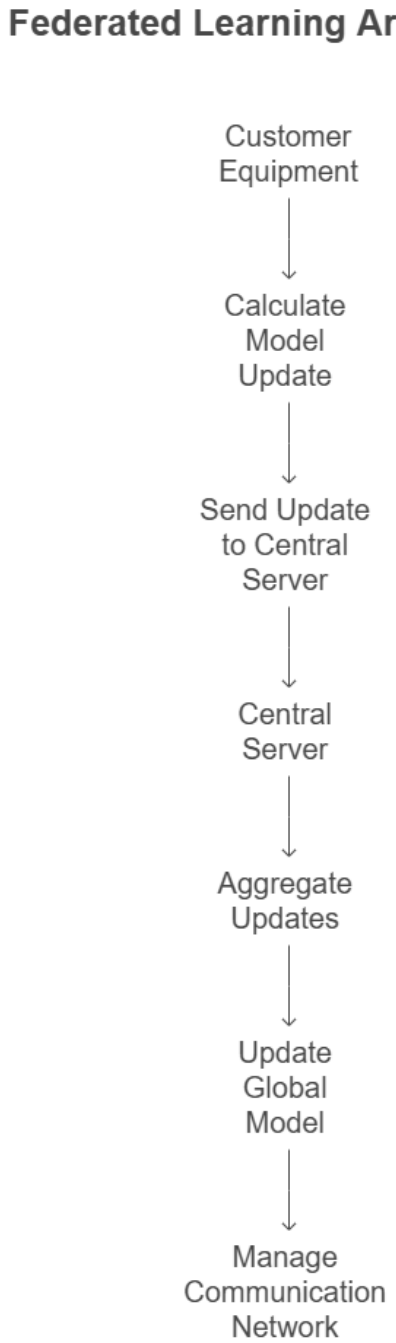

Figure 1. Basic FL architecture.

3.3 Basic Workflow in FL

The workflow of FL consists of several key steps, including initialization, local training on client devices, model updates and aggregation at a central server, and global model refinement. This cycle continues until the model reaches the desired level of accuracy or meets predefined stopping criteria. The Figure 2 elaborates the workflow model and below is the detailed explanation.

- **Initialization:** The central server integrates the global model with random or pre-trained parameters.
- **Local training:** Customers (Edge units) train models locally using their own data. They calculate the model grades or parameters.
- **Model updates and aggregation:** Customers send their updates to the Central Server, which collects the update using an algorithm as a FedAvg.
- **Global Model Update:** The update collected is used to update the global model, which is later sent back to the client unit for further training.
- **Repeat:** The process is repeated until the global model is convergence, or restriction criteria are completed.

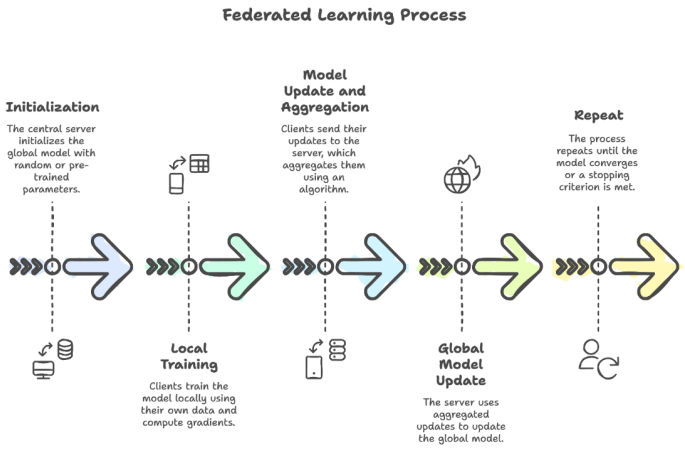

Figure 2. Workflow in FL.

The associated learning system provides a compelling solution for the AI model training, which provides privacy protection, low data transfer and decentralized processing. The development of FL has seen an increase in adoption in different fields, and architecture is sewn to accommodate the resource information environment as a built-in system and IoTs.

Built-in systems are widely used in various applications, including the Internet of Things (IoT), autonomous vehicles, health equipment and smart home automation. Traditional ML models require centralized data processing, where raw data is collected and sent to a central server for model training. However, this approach presents many challenges are shown in Table 1:

- **Privacy considerations:** Many built-in systems collect sensitive data (eg the health care system's monitoring devices, smart cameras). Dissemination of raw data on a centralized server caused significant privacy risk.
- **Bandwidth boundaries:** built-in devices often work in the atmosphere with limited network band width, making data transfer expensive and disabled.
- **Loadstones barriers:** real-time applications, such as autonomous vehicles and industrial automation, require low suppression treatment, which cannot always be guaranteed by centralized training.
- **Energy efficiency:** built-in systems usually have limited power resources. Sending large versions of data for processing can lead to rapid flow.
- **Scalability:** Increasing number of connected devices make centralized ML impractical, as it leads to data processing and storage [7].

FL addresses these challenges by allowing a built-in system to the Train ML model without broadcasting raw data. Instead, models are shared updates, retaining data privacy and adapting to the use of resources.

3.4 How FL Fits into the Resource Quality Environment

FL is designed to function effectively in the resource environment by taking advantage of the following strategies. Here's a Table 2 summarizing how FL fits into the resource-quality environment:

- **Decentralized learning:** FL enables built-in devices to train models locally and only transfer models' updates (gradients or parameters), which significantly reduces the data transfer requirements.
- **Model compression and optimization:** Techniques such as permineralization, pruning and knowledge distillation are used to reduce the size and calculation complexity of the model, which makes it possible for built-in systems.
- **Adaptive aggregation:** FL Employees Techniques such as Federated Averaging (FedAvg) to collect the model update effectively, reduce calculation and communication costs [21].
- **Downed FL architecture:** Instead of direct communication with a central server, hierarchical structures (eg edge-based aggregation) allow the mediated edge tools to collect updates before forwarding the cloud [7].
- **Energy-ability training:** Built-in systems can use techniques such as selective update transfer and event-controlled learning to adapt the power consumption.
- **Security and Privacy Mechanisms:** Federated Learning includes disruptive and secure aggregation methods to protect and reduce unfortunate attacks of sensitive data.

3.5 Applications of FL in Embedded Systems

Application of FL in built-in systems: Federated Learning has a wide range of applications in the built-in system, which enables smarter and safer AI-powered solutions. FL is increasingly being incorporated into practical embedded applications in a variety of domains with significant privacy, communication efficiency, and scalability benefits. Integration is extremely relevant to new and upcoming areas such as smart agriculture, edge-based medical diagnosis, and energy-conscious edge robotics. Here's a Table 3 summarizing the applications of FL in embedded systems: Some major applications include: Below is a step-by-step description of how FL supports these applications and the added value it provides to real-world edge AI use cases:

3.5.1 Smart Agriculture

In smart agriculture, FL has a key role as it enables collaborative ML in a decentralized network of distributed IoT sensors, drones, and farm equipment. The biggest challenges facing agriculture are how to handle huge volumes of real-time sensor data (e.g., crop health, temperature, and soil moisture) without breaching data confidentiality and reducing network bandwidth usage [21].

- FL keeps data in the edge devices (e.g., IoT sensors or drones) locally and transmits only model updates, thus protecting proprietary data.
- **Efficiency of Communication:** Instead of pushing all raw data to a master server, FL reduces bandwidth-hungry data communications. Models are trained locally and incremental updates are shipped, thereby saving communication resources, particularly in remote or off-grid farmland where there are few network facilities.

Table 1. Federated learning in embedded systems.

Aspect	Challenges in Traditional ML	How FL Addresses It
Privacy	Centralized data collection exposes sensitive user data.	FL keeps data localized, only sharing model updates.
Bandwidth	Limited network bandwidth makes large data transfers costly.	FL reduces data transmission by only sharing model parameters.
Latency	Real-time applications require low-latency processing.	Local model training ensures faster decision-making.
Energy Efficiency	Embedded systems have limited power for processing and transmission.	FL optimizes computation and minimizes communication.
Scalability	Centralized ML struggles with a growing number of devices.	FL enables decentralized learning across multiple devices.

Table 2. How FL fits into the resource-quality environment.

Strategy	Description
Decentralized Learning	FL allows devices to train models locally and only send model updates (gradients or parameters), reducing data transfer needs.
Model Compression & Optimization	Techniques like pruning, quantization, and knowledge distillation help minimize model size and computational complexity, making FL feasible for embedded systems.
Adaptive Aggregation	FL uses methods like Federated Averaging (FedAvg) to efficiently collect model updates while reducing computational and communication costs.
Hierarchical FL Architecture	Instead of direct communication with a central server, hierarchical structures (e.g., edge-based aggregation) allow intermediate edge devices to collect updates before sending them to the cloud.
Energy-Efficient Training	Embedded systems can optimize power consumption using techniques such as selective update transfer and event-triggered learning.
Security & Privacy Mechanisms	FL integrates methods like differential privacy and secure aggregation to protect sensitive data from exposure and adversarial attacks.

- **Personalization:** Every single farm can have its own setup of environment. FL enables devices to learn and train models locally under specific conditions and, therefore, generate personalized outputs for a specific farm, e.g., best irrigation timing or tailored pest control.
- **Scalability:** FL enables AI-powered analytics at scale. New sensors and devices deployed at various sites in the agriculture sector can both collaborate and improve models together without bogging down central systems [7].

3.5.2 Edge-Based Medical Diagnostics

In medicine, FL can transform edge-based medical diagnostics in revolutionary ways. This includes areas such as wearable health monitoring devices, medical imaging, and mobile health applications, where real-time, privacy-preserving insights are critical.

- **Data Privacy:** Healthcare information, such as patient health records, imaging information, and wearable sensor readings, is extremely sensitive. FL ensures that no such information ever leaves the patient's device or local healthcare organization, reducing privacy risks from central storage of data and complying with regulations such as HIPAA (Health Insurance Portability and Accountability Act).
- **Real-Time Decision-Making:** In medical diagnostics, real-time readings can be the difference in patient care. FL allows for local training of the model at the edge, and thus makes real-time processing of medical data, such as ECG signals, blood glucose, or X-ray images, without the wait of data transfer to central points.
- **Personalization:** Medical therapies may be tailored in multiple dimensions. FL enables health monitors and wearable sensors to tailor diagnostic

models according to the patient's individual data, resulting in customized medical treatments. For example, wearable sensors can continuously improve for the early identification of illnesses such as arrhythmias and diabetes problems.

- **Collaborative Learning:** Local hospitals and clinics could share a consolidated dataset. The information could be combined and used in tandem to train the diagnostic models together. It can be helpful to develop better predictions and models to diagnose patients without having to share the identity of each patient.

3.5.3 Energy-Aware Edge Robotics

FL is also contributing meaningfully in the area of edge robotics with energy efficiency, i.e., autonomous mobile robots, drones, and other robotic systems based on the edge implemented in manufacturing, logistics, and environmental monitoring use cases.

- **Energy Efficiency:** In-field robots would normally have constraints on their energy budgets, and sending a lot of data to central servers normally requires energy. FL allows robots to learn and update their model in the field with minimal communications and energy requirements. This benefits battery-powered units in the field, enabling them to operate in an efficient way for longer.

- **Collaborative Robotic Systems:** For use in areas like drone swarms or warehouse robots, FL can enable that multiple robots learn from their local environment as a group and exchange as little data as necessary. Every robot is capable of helping in model update, local experience learning independently, and adding to system knowledge without requiring data to be taken to a central point.

- **Local Adaptation to Environment:** Robots deployed in varying environments (diverse terrain, factory plant floors, or warehouses) may find themselves being exposed to local difficulties that would require local learning adaptation. FL facilitates adaptation at the local level but benefits from global knowledge acquired by all devices.

- **Security and Robustness:** FL lowers security risks of exposing edge devices to direct attack by malicious parties, i.e., model poisoning. By enabling local model updating and communication of aggregate updates instead of individual parameters, the system is less susceptible to intentional modification [42].

3.6 Smart Home Automation

FL is transforming smart home systems by enabling devices to learn user preferences regarding environmental control and security without violating personal data. Such devices, such as smart thermostats, lighting, and security devices, can adapt themselves based on local data without transmitting sensitive user information to central servers.

- **Personalization of User Preferences:** FL allows smart devices like smart thermostats, lighting systems, and security cameras to learn and adapt together based on individual user patterns and preferences. Training is local to the device, i.e., user-specific data (like favorite temperature, light levels, or security settings) is kept private to servers outside the device.

- **Voice Assistant and Smart Speakers:** Platforms like Google Home or Amazon Alexa can utilize FL to enhance voice recognition and natural language processing on their platforms. The framework can fine-tune user experiences in the local environment to enhance speech recognition, personalized responses, and recommendation systems without violating users' privacy and sending any data to cloud servers.

- **Energy Optimization Consumption:** FL can be used by smart meters and energy-efficient appliances to learn from real-time consumption to optimize maximum energy use in the home. FL can real-time optimize heating, cooling, lighting, and appliance controls based on usage to optimize energy efficiency without any impact on user data.

3.7 Autonomous Vehicles

In autonomous driving, FL enables the cars to learn from each other's experience without exchanging sensitive driving information with the cloud and improving the safety, performance, and efficiency of autonomous systems.

- **Navigation and Threat Alerts:** FL may be utilized by autonomous vehicles to share navigation route, road condition, and threat alert information without sharing raw data such as GPS coordinates, camera images, or driving habits. Decentralized learning enables cars to continuously update decision-making algorithms, improving safety features such as adaptive cruise

control, collision avoidance, and lane-keeping assistance.

- **Continuous Safety Enhancements:** Cars can also collaborate in real-time to improve their safety functions through the sharing of models that more accurately identify hazards, adaptive braking based on learning from conditions, and emergency response systems. FL makes it possible to carry out these upgrades locally on the vehicle without the necessity for the sharing of sensitive driving data with a central point.
- **Traffic Control and Route Guidance:** FL can be used by traffic control systems to dynamically control traffic in real time and provide advanced route guidance to vehicles. FL can avoid traffic congestion and synchronize traffic signal times using decentralized vehicle and traffic sensor information without violating the privacy of individual drivers.

3.8 Industrial IoT (IIOT)

FL for Industrial IoT allows production line machinery and factory devices to learn from local data to optimize processes, identify anomalies, and prevent equipment failure, without exposing proprietary production data.

- **Predictive Maintenance:** FL can be employed to update predictive maintenance models utilizing on-board sensors from machinery and equipment. The sensors monitor preliminary failure signals (e.g., vibration, temperature, pressure variations) and refresh the models without relaying unprocessed sensor data. This enables firms to maintain operations and minimize downtime by forecasting maintenance requirements prior to equipment failure.
- **Supply Chain Optimization:** FL can be employed to improve supply chain management, enabling diverse businesses to optimize operations, forecast demand, and manage stocks without accessing sensitive business information. Each plant can enhance the overall model's efficiency without revealing confidential operational data.

3.9 Intelligent Urban Development and Infrastructure

FL greatly enhances the development of intelligent and sustainable cities by facilitating decentralized intelligence for public safety, energy distribution, and urban transportation.

- **Public Surveillance Systems:** FL may augment public safety by enabling surveillance cameras and sensors to collaboratively learn to identify potential threats, security abnormalities, or suspicious activities. The system can enhance its detection models over time by utilizing local data from municipal cameras, without transmitting raw video feeds to a central server, thereby preserving privacy and security.
- **Energy Distribution Optimization:** FL can enhance energy use in smart cities by enabling various infrastructure components (e.g., smart grids, meters, and energy consumption devices) to develop localized models based on regional consumption trends.
- **Production Efficiency:** Smart companies can utilize FL to optimize their production efficiency. For example, factory equipment and floor sensors can collaborate to analyze operating efficiency, spot bottlenecks, and optimize the manufacturing process for increased output while protecting confidential and sensitive production information.

FL provides a promising solution for integrating ML into built-in systems by addressing privacy, bandwidth, delay and energy barriers. By taking advantage of decentralized learning, model adaptation and adaptive aggregation, FL enables effective and scalable AI-powered applications in the process-intensive environment.

4 Challenges, Research Gaps, and Future Directions

FL data enables decentralized training of models on different devices without giving up privacy. However, the implementation of FL on the built-in system presents many challenges such as the unit, security, data division and communication.

- **Heterogeneity of Devices**
  - **Calculation barriers:** Built-in systems usually have limited processing power, memory and energy resources, making it challenging to train complex models effectively.
  - **Venus's architecture:** Unit machine product layout (CPU, GPU, accelerator) can be different and therefore perform different performances and adjustments.

Table 3. Applications of FL in embedded systems.

Application	Description
Healthcare	• Wearable health devices (e.g., smartwatches, fitness trackers) train models to detect irregular heart rates, blood pressure deviations, and other health conditions while preserving patient data privacy. • Smart hospitals use FL to enhance disease spread models without sharing sensitive records.
Smart Home	• Home automation systems learn user preferences for lighting, temperature, and security without sending personal data to the cloud. • Voice assistants and smart speakers refine speech recognition models locally. FL also optimizes energy consumption by training models on local data from smart meters.
Autonomous Vehicles	• Self-driving cars share navigation patterns, road hazard detection, and recognition updates without exposing raw driving data. • FL improves safety features like adaptive cruise control and lane-keeping. • Traffic management systems use FL for real-time route optimization while preserving user location privacy.
Industrial IoT (IIoT)	• Predictive maintenance models are trained on embedded devices to detect equipment failures. • Smart factories analyze production efficiency with FL without exposing proprietary manufacturing data.
Agriculture & Environmental Monitoring	• FL enables precision agriculture by training crop monitoring models using distributed field sensors without centralized data storage. • Environmental monitoring systems improve pollution and weather forecasting models with minimal bandwidth usage. • FL supports sustainable farming by optimizing irrigation and resource management.
Smart Cities & Infrastructure	• Public surveillance systems enhance security threat detection without streaming raw video to cloud servers. • FL customizes energy distribution in smart grids by training decentralized models on local consumption data. • Traffic lights and road sensors collaborate to improve urban mobility through FL-powered traffic optimization.

  - **Software inequality:** Variability in the operating system, firmware versions and software libraries lead to difficulties in a standard FL structure.

- **Data Privacy and Security Problems**
  - **Model threat to inverted attacks:** Although FL does not share raw data, attackers can recreate sensitive information from model updates.
  - **Malicious participants:** Some equipment can be erected malicious, injecting biased or misleading updates and poisoning of the model.
  - **Encryption Overhead:** Homomorphic encryption and differential privacy Incur Calculation and ensure aggregation techniques such as calculation and communication costs, which can make built-in systems sick.
- **Model aggregation and non-IID data**

  - **Non-IID data division:** Introduced units collect data from the asymmetrical environment, causing global models to influence convergence-cracking and unbalanced datasets.
  - **Personal model adaptation:** A single global model cannot perform well in all devices due to separate data division, which requires individual FL approaches to connect the complexity.
  - **Unstable Convergence:** Variability in data distributions and computational capabilities can result in slow or unstable convergence of the federated model.

- **Communication overhead and efficiency**
  - **Limited bandwidth:** built-in devices are often dependent on low-bandwidth networks (eg IoT protocols such as MQT, LPWAN), causing constant model update.
  - **High energy consumption:** Modeled model updates often reduce battery life to resource-related equipment.
  - **Asynchronous partnership:** Equipment can go offline or stop connection, disrupt the training process and require techniques such as Asynchronous FL or client selection strategies.

- **Ultra-Low-Latency Constraints:** Edge devices, particularly for real-time applications like self-driving cars and industrial control, require ultra-low-latency computation. FL, while reducing the requirement to send data to the central servers, is still struggling to match the requirements of latency. Model updates and training must be performed rapidly without disrupting real-time decision-making. The necessary speed in communication between edge devices and the server and local processing capability is required to meet these standards.

- **Real-Time On-Device Learning:** For the majority of edge AI applications, models will need to learn and adapt at every moment. However, the ability to perform on-device real-time learning presents considerable challenge. The embedded devices usually have sparse processing capabilities and memory, making it hard to manage big data sets as well as perform recurrent model updates. Model pruning, quantization, and knowledge distillation are some of the methods employed for optimizing learning without compromising performance or energy efficiency.

- **Hardware Heterogeneity:** Edge devices are extremely heterogeneous in terms of their hardware capabilities, including different processors, memory, and power constraints. This hardware heterogeneity makes deployment of FL systems difficult because models need to be designed to operate well on a wide range of devices. Some devices may not be capable of executing certain computations involved in training models, which leads to difficulty in synchronization and ensuring successful cooperation among edge devices.

In order to remove these challenges, adapted FL frameworks must take into account lighter model architecture, effective encryption techniques, adaptive aggregation methods and correlation-qualified strategies for built-in systems.

5 Techniques and Frameworks Enabling FL in Embedded Systems

To address the problem of FL in embedded systems, numerous methods have developed that attempt to improve efficiency, security, and scalability. Such methods enhance model training based on the computational and communication constraints of embedded devices.

5.1 FL Algorithms for Embedded Systems: FedAvg and FedProx with Real-World Applications

Federated Averaging (FedAvg) is the cornerstone algorithm for federated learning (FL). In FedAvg, local devices independently educate models the use of their personal information and percentage only version updates with a valuable server for aggregation. Mathematically, if each device k has n_k data points, and w_k is the nearby version, the server updates the worldwide version w by using weighted averaging:

$$w \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k \quad \text{where} \quad n = \sum_{k=1}^{K} n_k \tag{1}$$

FedAvg has been practically carried out in numerous embedded environments. A tremendous instance is Google's Gboard keyboard gadget. In this machine, consumer smartphones embedded gadgets with limited processing and reminiscence assets collaboratively train subsequent-word prediction

fashions without transmitting non-public typing data to imperative servers. Each phone regionally fine-tunes the language model using user interactions, then sends encrypted model updates to the server. This guarantees low conversation overhead while keeping person privateness — a critical consideration for embedded cell systems running in diverse network environments. In the healthcare domain, FedAvg was used to educate models throughout wearable medical devices used for monitoring cardiac activity. Each wearable device, like a heart fee sensor, amassed personal fitness information and skilled a nearby version to detect early signs of cardiovascular anomalies. The local updates, in place of the sensitive raw fitness records, were aggregated centrally. This FL-based totally approach included affected person privateness at the same time as allowing the advent of correct, population-wide predictive health fashions, demonstrating how FedAvg fits clinical embedded structures limited with the aid of privateness and battery life.

However, embedded systems often gift demanding situations which includes hardware heterogeneity and records non-IIDness (non-Independent and Identically Distributed facts), which could have an effect on FedAvg's performance. To deal with those troubles, FedProx modifies FedAvg's local schooling goal through adding a proximal term that penalizes deviation from the global version, mathematically:

$$\min_{w} f_k(w) + \frac{\mu}{2}\|w - w_{\text{global}}\|^2 \tag{2}$$

where μ controls the regularization strength. The proximal term allows stabilize schooling when gadgets have differing computational abilities or whilst statistics distributions are extraordinarily skewed. In a clever domestic case have a look at via [15], FedProx turned into used to train models across diverse embedded devices like smart thermostats, protection cameras, and lighting fixtures structures. These gadgets had relatively personalized usage styles and ranging computational assets. By applying FedProx, researchers ensured that neighborhood fashions educated successfully no matter heterogeneity in device electricity and person conduct. The ensuing global model enabled extra intelligent and customized electricity control systems without compromising person data privateness or requiring regular cloud connectivity.

Another sturdy utility of FedProx became demonstrated within the business IoT (IIoT) environment with the aid of. Manufacturing gadgets prepared with embedded sensors varied extensively in phrases of records generation costs and hardware talents. By the use of FedProx, predictive upkeep models were trained throughout those distributed systems to hit upon gadget disasters early. The proximity constraint avoided devices with noisy or low-quantity facts from negatively impacting the global model, enhancing both the accuracy and stability of the federated gadget. This reduced downtime in business operations and showed how FL can be a game-changer for smart factories.

In the context of self-reliant cars, [35] explored making use of FL to onboard embedded processors. FedAvg was to begin with used to enable automobiles to collaboratively teach object recognition and lane detection fashions without uploading raw sensor statistics to a cloud. However, due to variations in riding environments (e.G., town traffic as opposed to rural roads), non-IID information became a vast hurdle. Introducing FedProx allowed each vehicle to conform education at the same time as retaining alignment with a global fleet-extensive model, improving robustness and decreasing version glide — crucial for preserving protection and overall performance in self-reliant driving systems.

Thus, each FedAvg and FedProx have validated instrumental in permitting federated studying across embedded structures. FedAvg is efficient when devices are exceedingly homogeneous, even as FedProx excels in environments with large device and facts heterogeneity, that is common in actual-global embedded programs. Their a success deployment in clever houses, healthcare wearables, self-reliant cars, and IIoT validate the sensible capability of federated gaining knowledge of to revolutionize AI deployment at the network edge even as upholding strict privacy, latency, and strength-performance necessities.

5.2 Model Compression and Perishable

Built-in units have limited memory and data capacity, and therefore FL is necessary for model compaction and finance for finance.

- **Weight Prize:** Removes fruitless parameters and weight from the nerve network to reduce the size of the model without damaging the accuracy.
- **Privation:** The model changes the model parameters in little perfect (eg 8-bit integer) than highs perception (eg 32-bit floating point),

reduces memory and calculation expenses significantly.

- **Knowledge distillation:** A small "teacher" trained by a large "teacher" model benefits from a small "student" model, maintains performance, but reduces calculation requirements.
- **Low rank constitutions:** Weight matrix is a factor in the form of low-ranking forms to reduce parameters while maintaining important functions.

5.3 Gradient Sparsification Techniques

considering that recurrent update transmission of a model is expensive for embedded hardware, gradient Sparsification is used to alleviate communication overhead.

- **Top-K Sparsification:** The most important shield sends the top K% of the update, which occupies the minimum bandwidth.
- **Random to drop:** A random selection of gradients that are useful in low bandwidth environmental magazines.

5.4 Secure Aggregation

A data adopts security methods, including improvement in safety against leaks, flies in built-in systems and safe aggregation.

- **Differential Privacy (DP):** The model introduces noise control for the update before sending the server, preventing the end of personal data
- **Secure Multi-Party Computation (SMPC):** Enables the device to calculate the collected model updates without highlighting personal data and secure privacy.
- **Homomorphic Encryption:** Encrypt model updates are such that they can be collected without decrypting to secure the data privacy.
- **Blockchain-based Safe FL:** User Blockchain Technology to tamper-illustrated logging and distributes the model updates, increases the security and confidence.

FL can be implemented with success on built-in systems, and combines federated optimization, model compression, gradient savings and safe aggregation techniques. Such methods reduce resource limits, increasing safety and communication reduces overheads, which makes FL -er more convenient for real-world applications in IoT, Smart Healthcare and Edge AI.

Federated Learning (FL) in built-in systems enables decentralized AI treatment while retaining privacy and reducing communication costs. Below are the most important applications where FL built-in AI changes the deployment. Applications of FL in embedded systems are clearly shown in Figure 3.

- **Health care and medical equipment**
  - **Personal health monitoring:** Use FL to analyze user health data (heart rate, sleep patterns) without sharing raw data (eg Smartwatch, Fitness Trackers), protect privacy.
  - **Medical imaging and diagnostics:** FL helps hospitals work on AI-based diagnostics (X-rays, MRI analysis) without dividing the patient's data into institutions.
  - **Glucose monitoring and insulin pumps:** built-in FL models in continuous glucose monitors (CGM) improve future accuracy for diabetes treatment.
  - **Prediction of the disease:** FL enables AI training to associate patient health records in many hospitals to detect initial illness (eg. heart disease, Parkinson's).
- **Autonomous Systems and Smart Vehicles**
  - **Autonomous driving:** FL allows to learn vehicles connected to real driving conditions without postponing sensitive location and behavioral data.
  - **Traffic flow optimization:** Smart traffic systems Use FL-competent built-in devices to analyze vehicle movement patterns and adjust the traffic signal dynamically.
  - **Driver behavior analysis:** AI systems in vehicles use FL to adapt help companies and secure privacy.
  - **Conflict prevention system:** built-in sensors and cameras in vehicles train Corporately AI models to detect real-time objects and predict danger.
- **Smart Homes and IoT Applications**
  - **Voice assistant and home automation:** Amazon Alexa and Google Home can

increase voice recognition locally without sending sensitive data to shuttle servers.

- **Energy handling:** Smart meters and thermostats use FL to optimize energy consumption patterns, and preserve domestic data privacy.
- **Discover deviations in smart homes:** FL helps to detect unusual activities in safety cameras or smart door locks without broadcasting sensitive recordings.
- **Personal AI Services:** FL enables smart -TVs and home entertainment systems to recommend material based on the pattern of use without central data collection.

- **Industrial IoT and Predictive Maintenance**
  - **Fault Detection in Manufacturing:** Detects in production: Factory lets the built-in sensors in the machinery predict the need for maintenance without learning from local errors and sharing sensitive production data.
  - **Optimization of the supply chain:** FL models in logistics networks optimize inventory and delivery routes based on real-time data from different places.
  - **Energy-capable factory operations:** Smart networks and industrial energy management system FL is used to balance load and prevent overcoming.
  - **Cyber security for Industrial IoT:** FL detects danger in the industrial network by collaborating in detecting training deviations in units distributed equipment.

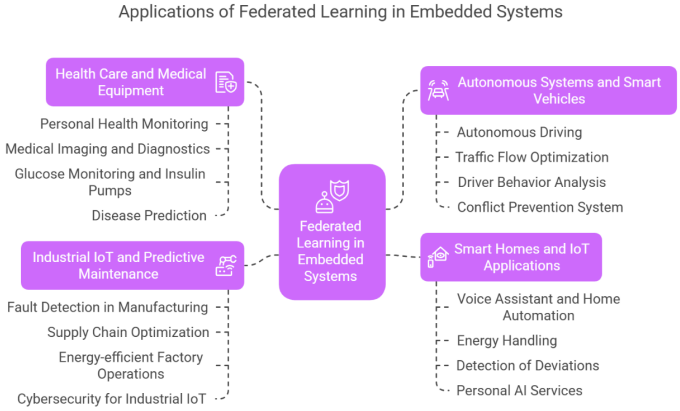

Figure 3. Applications of FL in embedded systems.

In the built-in system, Federated Learning revolutionizes Healthcare, Autonomous System, Smart Homes and Industrial IoT AI in different fields. By activating decentralized model training, FL data increases security, privatization and efficiency by reducing transfer costs.

Recent advancements in FL highlight the incorporation of lightweight models, energy-efficient hardware, and reinforcement learning to improve scalability and performance in embedded and edge AI systems. These nascent study domains are essential for facilitating the practical implementation of FL in resource-limited settings. Principal research trajectories encompass:

5.5 TinyML for Low-Power Embedded Systems

TinyML focuses on deploying AI models on ultra-low-power embedded devices, such as microcontrollers and sensors. The advancements are

- **Ultra-Lightweight Models:** Neural network architectures like MobileNet, SqueezeNet, and EdgeTPU-optimized models allow efficient on-device training.
- **Hardware-Accelerated Learning:** Specialized AI chips (e.g., Google Coral, NVIDIA Jetson, Arm Cortex-M) enable energy-efficient FL on edge devices.
- **Sparse and Quantized Models:** Techniques like pruning, weight sharing, and quantization reduce model size, making FL feasible on constrained devices.
- **On-Device Continual Learning:** Embedded FL adapts models over time without requiring cloud-based retraining, improving personalization.

Research Focus: TinyML's combination with FL aims at on-device learning with minimal data being sent to the cloud, which is suitable for use in remote locations with constrained connectivity [11]. Notable areas of research explore the development of highly compact models that can be run on devices with only a few kilobytes of memory, and model quantization and pruning methods for model size and computation reduction.

5.6 FRL

FRL extends FL to reinforcement learning (RL) problems, enabling decentralized agents to learn optimal policies collaboratively. FRL is the combination of FL and RL concepts to provide distributed learning in situations where agents (e.g., drones, autonomous vehicles) can learn from feedback

from their environment and experience shared among group members without compromising privacy [38].

- **Autonomous Systems:** Smart robots, drones, and self-driving cars use FRL to learn navigation strategies without central data collection.
- **IoT Device Coordination:** Smart grids, industrial automation, and collaborative robots (cobots) optimize energy use and task scheduling
- **Personalized Edge AI:** Wearables and healthcare devices use FRL to customize user interactions without exposing private data
- **Multi-Agent Learning:** Distributed agents in smart environments (e.g., traffic systems, disaster response) coordinate decisions while preserving privacy.

Research Directions: FRL allows multiple independent agents to improve their decision models (e.g., path planning, collision avoidance) collaboratively through sharing updates in a privacy-preserving manner. Research includes decentralized agent coordination, reward sharing, and policy update that is exploration-exploitation balanced within a federated setting.

5.7 Integration of Edge Computing with FL

Edge computing reduces reliance on cloud infrastructure by processing data closer to the source, minimizing latency and bandwidth usage. The key innovations are

- **Hierarchical FL (HFL):** Edge nodes aggregate local model updates before sending them to the central server, reducing communication overhead.
- **On-Device Inference & Training:** Embedded AI models perform both inference and incremental updates locally, making FL more efficient.
- **Edge AI Hardware:** New processors (e.g., Google Edge TPU, Intel Movidius, Raspberry Pi 5) accelerate FL workloads in resource-constrained environments.
- **Adaptive Client Selection:** Dynamic edge-based FL systems select only relevant devices for training, optimizing resource utilization.

Research Focus: FL integration with low-power accelerators is focusing on increasing deep learning computations' energy efficiency [1]. Researchers are attempting to fine-tune hardware-centric features (e.g., low-precision computing, quantization) so they provide the ideal tradeoff among power consumption and model accuracy for federated training [16].

5.8 Blockchain and FL for Security

Blockchain technology ensures secure, transparent, and tamper-proof model aggregation without centralized trust. The recent developments are

- **Decentralized FL:** Eliminates the need for a central server by using blockchain for model aggregation, reducing single points of failure.
- **Smart Contracts for Incentives:** Devices contributing high-quality updates can be rewarded using blockchain-based token systems.
- **Secure Multi-Party Computation (SMPC) + Blockchain:** Combines cryptographic techniques with distributed ledgers for enhanced privacy and security.
- **Federated AI Marketplaces:** Blockchain enables privacy-preserving data marketplaces where devices can share model updates securely.

Research Focus: Blockchain and FL is an efficient solution to security and privacy concerns in decentralized AI systems. Blockchain enables secure and transparent aggregation of models without centralized trust. Decentralized FL, where model update is controlled by the blockchain in a decentralized manner, reducing the points of failure to one is some of the recent work. Furthermore, smart contracts facilitate quality input from devices through a token-based reward system effectively [40]. Blockchain also safeguards privacy through Secure Multi-Party Computation (SMPC) with cryptographic techniques and distributed ledgers for secure data aggregation [22]. Federated AI marketplaces based on blockchain also facilitate secure sharing of model updates with preserving data privacy and allowing a decentralized data exchange [23]. These developments mark a significant achievement in the development of secure, privacy-preserving, and decentralized AI technology for embedded systems.

The convergence of TinyML, FRL, Edge Computing, and Blockchain is shaping the future of FL in embedded systems. These advancements enable more efficient, secure, and intelligent AI solutions across diverse applications, from IoT and healthcare to autonomous systems and industrial automation.

6 Security and Privacy in FL for Embedded Systems

FL facilitates decentralized model training with data stored on the local devices. However, security and privacy attacks such as data leakage, adversarial attacks, and tampering of models are still important concerns. Some of the most significant techniques to enhance privacy and security in FL are explained below.

6.1 Privacy Protection Technique:

Privacy Protection Mechanisms ensure that FL participants do not leak sensitive information when dividing the model updates.

- **Differential Privacy (DP):** The model provides controlled noise to the model grades before transferring, and prevents the attackers from mentioning individual data points. DP-fed hob Privacy Conservation Model Changes the traditional association average with noise injections for updates. Model at high privacy level can reduce accuracy.
- **Homomorphic Encryption:** This model update can therefore be calculated (eg yoga) data without decrypting. Parlier encryption allows encrypted model aggregation without highlighting individual updates. Calculation overhead increases, making it less convenient for resource-limited units.
- **Safe aggregation:** FL servers receive the collected model updates instead of individual contributions, which prevent exposure to data on a device. Eg: The safe aggregation of Google hides the individual model update by encrypting them before collecting protocol. Local Different Privacy (LDP) Uses direct noise at unit level before transfer, also ensures privacy in small-scale FL layout [7].

6.2 Safe Multi-Party Computation (SMPC):

SMPC enables many FL participants to calculate a function on their personal information.

- **Secret sharing technique:** Each model update is divided into several shares distributed among different parties. Only when joint updates can only be rebuilt. The secret sharing of Shameer shares the model gradients between multiple units to increase privacy [21].
- **Circuit made:** Allows the device to safely perform functions without revealing input. Privacy protection is used for federated medical analysis.
- **Decentralized federal learning with SMPC:** Exits the central aggregation server by allowing units to safely communicate using the SMPC protocols. Calculation and communication costs increased. Bag Protocol Design is required to balance efficiency and privacy.
- **Adversarial Attacks and Robustness of FL:** FL is vulnerable to various attacks that can compromise model integrity and privacy.
- **Model Poisoning Attacks:** Malicious participants inject manipulated model updates to degrade global model performance. Byzantine-resilient aggregation (e.g., Krum, Trimmed Mean) filters out abnormal updates.
  - **Data Poisoning Attacks:** Attackers modify local training data to introduce biases in the global model. Robust FL techniques like anomaly detection and outlier rejection help mitigate this.
  - **Inference Attacks (Model Inversion & Membership Inference):** Adversaries attempt to reconstruct private training data from shared model updates. Differential privacy and gradient obfuscation techniques reduce data exposure.
  - **Evasion Attacks (Adversarial Examples):** Attackers craft input data that misleads the trained model (e.g., causing misclassification in image recognition). Adversarial training and defensive distillation enhance model robustness.

Ensuring security and privacy in FL requires a multi-layered approach combining differential privacy, encryption, SMPC, and adversarial defense mechanisms. Ongoing research focuses on reducing computational overhead while strengthening FL's resistance to attacks, making it more reliable for real-world applications.

6.3 Model poisoning, data poisoning, and adversarial attack

Ensuring safety and privateness in FL is crucial, in particular in embedded systems, wherein sensitive records from healthcare devices, autonomous automobiles, and IoT sensors ought to be safeguarded. Although FL inherently reduces the danger of facts

leakage by using maintaining raw records localized, it is nevertheless susceptible to state-of-the-art assaults along with model poisoning, records poisoning, and adversarial assaults.

- **Model poisoning** attacks involve malicious clients intentionally sending manipulated version updates to deprave the global model. For instance, an attacker should inject updates that degrade model accuracy or introduce unique backdoors. A well-known case consists of "backdoor assaults" in which an adversary subtly poisons a fragment of the clients to make the version misclassify unique inputs whilst keeping high accuracy on smooth statistics [4].
- **Data poisoning** assaults goal the nearby datasets on purchaser devices. Here, malicious records points are inserted into the schooling records to mislead the model. In useful resource-limited embedded environments like smart sensors, detecting and mitigating facts poisoning is mainly tough because of restricted computational abilities [37].
- **Adversarial attacks** similarly exacerbate FL vulnerabilities by way of crafting diffused input perturbations that idiot the model into wrong predictions. Attackers might not even need get right of entry to to the schooling facts without delay but can infer facts from the updates shared during FL rounds [18].

To combat those safety threats, numerous mitigation strategies have been proposed:

- **Robust aggregation techniques** consisting of Krum, Trimmed Mean, and Bulyan make sure that malicious customer updates have minimal impact on the worldwide model by using selectively aggregating handiest dependable updates [5].
- **Differential privacy (DP) mechanisms**, by using including calibrated noise to model updates, shield against leakage of sensitive information from the shared parameters while keeping a reasonable stability between privacy and model overall performance.
- **Secure Multi-Party Computation (SMPC) and Homomorphic Encryption (HE)** permit version updates to be aggregated without revealing the individual updates, as a consequence retaining customer privacy even for the duration of communication [6].
- **Anomaly detection** at the server facet is hired to perceive abnormal updates, that could symbolize a poisoning attempt. Thresholding strategies and clustering analysis are sensible for lightweight embedded structures [33].
- Recent advances have also added **Byzantine-resilient optimization algorithms** that tolerate the presence of a positive fraction of antagonistic customers with out significantly impacting the convergence or accuracy of the model [9].

In embedded systems, wherein gadgets are often heterogeneous and resource-confined, light-weight protection solutions tailor-made for federated settings are vital. Future paintings ought to awareness on growing adaptive protection frameworks that bear in mind computation, verbal exchange, and strength constraints whilst making sure resilience against a huge range of opposed threats.

FL has shown significant opportunities in AI protection, but many challenges remain. Future scalability is focused on improving scalability, handling unit's asymmetry, reducing communication and calculations overhead and taking advantage of the next generation network such as 5G and beyond [24].

6.4 Scalability and adaptability in FL

Since flatter to spread to billions of age units, it is an important challenge to ensure effective model training on large-scale networks. Major challenges are

- **Scalability in large networks:** As the number of participants increases, administration of the model updates and the collection becomes more complicated than effective.
- **Dynamic participation:** Units often join in and leave the FL network, and require adaptive training mechanisms.
- **Federation hypermeter optimization:** It is difficult to adapt to learning speeds, batch sizes and updated frequencies in different devices.

Future solutions:

- **Hierarchal FL (HFL):** Before sending to the central server, use intermediate age nodes to collect updates before reducing the load and improving scalability.
- **Self-adaptive FL model:** Algorithm that adjusts exercise parameters dynamically based on device

skills and network conditions.

- **Decentralized FL architecture:** Blockchain-based or colleague to move away from learning point of view, away from centralized aggregation.

6.5 Handling device with heterogeneity

FL includes different hardware functions, operating systems and different types of devices with network conditions. Major challenges are

- **Calculation variability:** Produced-divis often contains limited memory, processing power and battery life.
- **Various network conditions:** Some devices may have high speed connections, while others depend on low bandwidth networks.
- **Non-IID data division:** Tools generate very individual and non-human data, which affect the model's convergence.

Future solutions:

- **Individual Federated Learning (PFL):** Development of customer-specific models instead of a single global model to accommodate device variability.
- **Asynchronous FL:** Let the device update the model at its own pace instead of following the synchronized training round.
- **Federated Transfer Learning (FTL):** Use knowledge transfer techniques to make global models compatible with specific device conditions.

6.6 Emerging solutions for communication and calculation costs

Communication efficiency is one of the largest hedges in FL, specially built-in and IoT units. Major challenges are

- **High bandwidth consumption:** Continuous model update leads to rush.
- **Energy deficiency:** built-in system has a limited battery life, which makes FL participation impractical.
- **Researcher problems:** Real-time FL-application (eg autonomous vehicles) requires low delay updates.

Future solutions:

- **Gradient Compression and Sparsification:** Techniques such as top-key savings and quantization reduce data transfer.
- **Adaptable client selection:** Choose only one of the optimal networks and calculation resources for each training round.
- **Federated distillation:** Instead of sharing full model updates, units exchange knowledge in a distilled form, reduces the communication load.
- **Edge-Cloud Collaboration:** Use hybrid architecture where the Edge devices make initial calculation before unloading the works on the cloud.
- **Edge-Cloud Collaboration:** Using hybrid architectures where edge devices perform initial computations before offloading tasks to the cloud.

6.7 Integration with 5G and beyond

The deployment of FL in embedded systems has been significantly greater by rising technologies such as 5G networks and side computing. 5G gives extremely-dependable low-latency conversation (URLLC) and huge system-type verbal exchange (mMTC), allowing speedy, real-time synchronization of model updates throughout heaps of heterogeneous embedded gadgets [34]. This is particularly beneficial in use instances consisting of self-reliant vehicles, where decentralized schooling improves riding fashions based on neighborhood reports without transmitting touchy sensor statistics to centralized servers. Similarly, area computing introduces localized aggregation and version validation, lowering verbal exchange overhead and latency through processing updates towards the statistics assets. In smart production settings, embedded sensors and robots collaboratively educate FL models to locate equipment faults and optimize operations without exposing proprietary business data to external servers [28].

Practical applications combining FL, 5G, and area computing encompass healthcare wearables, clever metropolis infrastructure, and precision agriculture. Devices like smartwatches use FL for anomaly detection even as benefiting from 5G-enabled rapid model updates [20]. In intelligent traffic systems, edge servers positioned at intersections combination updates from vehicular sensors to optimize site visitors go with the flow even as maintaining driving force privacy [31]. Moreover, customized content material shipping on cellular systems, such as Google's Gboard,

leverages FL to enhance predictive typing without user statistics leaving the device. Emerging side-cloud collaborations are also being utilized in agriculture, permitting distributed sensors to predict soil and climate conditions collaboratively. As 5G and aspect infrastructure extend globally, the destiny of FL in embedded systems factors toward fantastically scalable, secure, and low-latency deployments across crucial industries. FL is expected to replace FL by activating the next generation network such as 5G and future 6G, high speed, low-hearted sign. Big benefits are

- **Ultra-Low latency:** Affairs of the important following model updates that are important for autonomous systems and real-time AI applications.
- **High bandwidth:** The network continuously allows data exchange without strong infrastructure.
- **Network slices:** By improving reliability, FL awards supplied network resources to applications.

Future solutions:

- **More than FL 6G and Quantum Network:** Use AI-driven, self-help network for real-time FL updates.
- **AI-enhanced 5G Edge Computing:** Combination of FL with edge calculation in 5G base stations for real-time treatment.
- **Federated Network Intelligence:** 5G-SAP FL to increase network traffic adaptation, security and increase unit-to-to-to-unit.

Here's a structured Table 4 summarizing the challenges and future solutions of FL in AI protection.

FL in AI protection faces several challenges and potential future solutions. FL's future depends on addressing scalability, heterogeneity, communication costs and integrating next-to-and-network. Edge data processing, decentralized architecture, adaptive learning and progress in 5G/6G will play an important role in making flick more efficient and was widely used in built-in and IoT systems.

Despite significant development in federated gaining knowledge of FL for embedded structures, several crucial studies gaps stay, specifically regarding model aggregation under heterogeneous conditions and sturdy privateness protection strategies.

6.8 Model Aggregation in Heterogeneous Environments

Traditional aggregation strategies like FedAvg assume that each one collaborating gadgets own identical version architectures, computational resources, and relatively balanced statistics distributions. However, in actual-international embedded structures, heterogeneity is the norm devices fluctuate significantly in hardware capabilities, information quality, availability, and reliability. This poses a couple of demanding situations:

- **Non-IID Data Bias:** Aggregating updates from devices with massively exceptional local statistics distributions can lead to biased international models, sluggish convergence, or even version divergence.
- **Partial Participation:** Due to confined energy, connectivity issues, or dynamic availability, best a subset of devices can participate in each spherical, making aggregation techniques designed for complete participation sub-finest.
- **Diverse Model Architectures:** In some embedded settings, devices may not even proportion identical version systems (e.G., light-weight CNNs on wearables vs. Deeper models on part servers), growing a need for heterogeneous version fusion techniques.

Current studies efforts inclusive of FedProx [25] and MOON (Model-Contrastive de introduced changes to deal with heterogeneity, but they regularly assume solid device competencies and nonetheless require synchronized replace durations. Future paintings have to focus on adaptive aggregation algorithms that dynamically weigh contributions primarily based on tool reliability, version nice, or context-aware metrics.

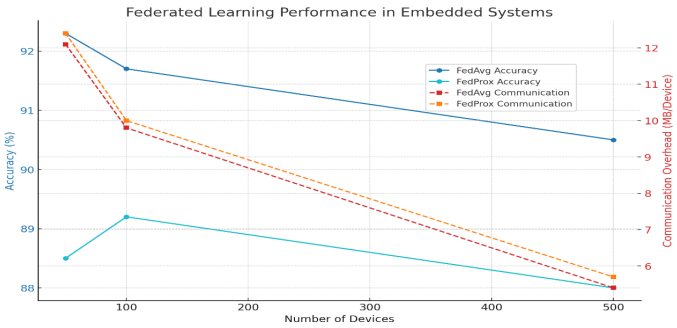

Figure 4. FL performance in embedded systems.

Table 4. Challenges and future solutions of FL in AI protection.

Category	Challenges	Future Solutions
Scalability & Adaptability	• **Scalability in large networks:** Managing model updates becomes complex as participants increase. • **Dynamic participation:** Devices frequently join and leave the FL network. • **Federation hyperparameter optimization:** Adapting learning speeds, batch sizes, and update frequencies across different devices is difficult.	• **Hierarchical FL (HFL):** Use edge nodes to collect updates before sending to the central server. • **Self-adaptive FL model:** Algorithms that dynamically adjust training parameters based on device and network conditions. • **Decentralized FL architecture:** Blockchain-based peer-to-peer learning instead of centralized aggregation.
Handling Device Heterogeneity	• **Computational variability:** Devices have different memory, processing power, and battery life. • **Varying network conditions:** Some devices use high-speed connections, while others rely on low-bandwidth networks. • **Non-IID data distribution:** Device-generated data is highly individual, affecting model convergence.	• **Personalized FL (PFL):** Develop client-specific models instead of a single global model. • **Asynchronous FL:** Devices update models at their own pace instead of following synchronized training rounds. • **Federated Transfer Learning (FTL):** Use knowledge transfer techniques to make global models adaptable to specific devices.
Reducing Communication & Computation Costs	• **High bandwidth consumption:** Frequent model updates cause network congestion. • **Energy inefficiency:** Embedded systems have limited battery life, making FL impractical. • **Real-time FL challenges:** Applications like autonomous vehicles require low-latency updates.	• **Gradient compression & sparsification:** Techniques like top-k selection and quantization reduce data transfer. • **Adaptive client selection:** Choose devices with optimal network and computational resources for training rounds. • **Federated distillation:** Devices exchange distilled knowledge instead of full model updates. • **Edge-Cloud Collaboration:** Hybrid architecture where edge devices perform initial computation before offloading tasks to the cloud.
Integration with 5G & Beyond	• **Ultra-low latency:** Critical for real-time AI applications like autonomous systems. • **High bandwidth requirements:** Ensures continuous data exchange with minimal infrastructure constraints. • **Network slicing:** Improves reliability by allocating dedicated network resources for FL applications.	• **FL over 6G & Quantum Networks:** AI-driven self-adaptive networks for real-time FL updates. • **AI-enhanced 5G Edge Computing:** Combine FL with edge computing in 5G base stations for real-time processing. • **Federated Network Intelligence:** Use 5G-enabled FL to enhance network traffic adaptation and security.

6.9 Privacy Protection below Heterogeneous Conditions

While differential privacy (DP) and stable aggregation protocols offer strong theoretical ensures, making use of them effectively in embedded settings stays an open hassle:

- **Resource Constraints:** DP noise addition or steady multi-celebration computation (SMPC) schemes are computationally luxurious and memory-in depth, that could overwhelm useful resource-confined devices which includes microcontrollers and IoT sensors.

Table 5. FL performance in embedded systems.

Devices	Data Distribution	Aggregation Algorithm	Accuracy (%)	Training Time (h)	Comm. Overhead (MB/device)
50	IID	FedAvg	92.3	2.4	12.1
50	Non-IID	FedAvg	88.5	2.8	12.4
100	IID	FedProx	91.7	4.1	9.8
100	Non-IID	FedProx	89.2	4.6	10.0
500	IID	FedAvg	90.5	10.2	5.4
500	Non-IID	FedProx	88.0	11.7	5.7

- **Gradient Leakage in Non-IID Settings:** In heterogeneous datasets, version updates can inadvertently leak more sensitive information compared to IID settings, because the updates become extra personalized and much less generalizable, making them less complicated to deduce.
- **Adaptive Adversaries:** Emerging assault fashions, along with adaptive membership inference assaults or version inversion attacks, goal the weaknesses of privateness mechanisms when devices have choppy defenses or numerous statistics sensitivities.

Although approaches which includes FedSGD with secure aggregation [7] and differentially personal FL frameworks were proposed, they often assume homogeneous device abilities or constant privacy budgets. Future directions should explore useful resource-conscious privacy techniques, tool-particular privacy budgets, and contextual differential privateness which could dynamically modify to each device's sensitivity and constraints. In addition, the mixing of hardware-primarily based security features which includes Trusted Execution Environments (TEEs) into FL for embedded systems remains in large part unexplored beyond remoted proofs-of-concept. Embedding light-weight, depended on computation zones at the tool stage should bridge the gap among robust privacy ensures and real-time embedded AI needs.

To examine how federated gaining knowledge of scales in embedded environments, a synthetic simulation was performed varying the variety of taking part devices, records distributions (IID vs Non-IID), and the aggregation algorithms (FedAvg and FedProx). Results indicate that whilst accuracy remains notably high as tool counts grow, non-IID records distributions lead to a modest overall performance drop, specifically for primary aggregation methods like FedAvg. FedProx always outperforms FedAvg beneath non-IID settings, even though at the fee of slightly accelerated training time. Communication overhead in keeping with device decreases with large networks, highlighting the scalability benefit of FL in bandwidth-limited embedded structures (see Figure 4). The values are explained in the Table 5.

Here's the output graph showcasing the synthetic simulation results for FL in embedded structures. It presentations both accuracy and conversation overhead for the FedAvg and FedProx algorithms across different numbers of devices (50, 100, 500).

7 Conclusion

Summary of Key Findings: FL is a promising privacy-preserving and decentralized technology for AI in embedded systems. The key findings present the main challenges, including device heterogeneity, communication overheads, and security and privacy attacks. However, FL is growing fast with TinyML, FRL, and blockchain. The advancements will be directed to reduce computation cost and communication cost, model performance and security. Furthermore, techniques like differential privacy and secure multi-party computation (SMPC) are addressing data confidentiality and model integrity challenges.

Future Research and Development Potential: There are a number of promising fields for future development of FL. In the future, future research can be directed towards improving the scalability of FL systems, especially for large-scale applications, through hierarchical FL and decentralized aggregation techniques. There is also room for personalized FL models, which can learn according to the unique capabilities of the devices they run on. Improving energy efficiency and reducing computation loads

will be crucial to allowing more extensive application of FL on low-resource devices. Furthermore, as 5G and subsequent networks become available, utilizing these for high-speed and low-latency communication will be key to facilitating real-time FL applications.

Impact on Real-World Applications: FL is already contributing immensely to several fields with effective, secure, and smart AI-powered solutions. In medicine, it is enabling the creation of privacy-preserving diagnostic devices, patient tracking, and personalized therapy without exposing sensitive medical information. In autonomous systems such as autonomous cars and drones, FL is enabling FL to achieve real-time decision-making without breaching data confidentiality. IoT technology, smart cities, and smart homes employ FL for energy management, device coordination, and service personalization. With the growth, it threatens to overwhelm markets such as industrial IoT, manufacturing, and supply chain optimization.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318). [Crossref]

[2] Abdelmoniem, A. M., Ho, C. Y., Papageorgiou, P., & Canini, M. (2023). A comprehensive empirical study of heterogeneity in federated learning. *IEEE Internet of Things Journal, 10*(16), 14071-14083. [Crossref]

[3] Hazra, A., Adhikari, M., Nandy, S., Doulani, K., & Menon, V. G. (2022). Federated-learning-aided next-generation edge networks for intelligent services. *IEEE Network, 36*(3), 56-64. [Crossref]

[4] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020, June). How to backdoor federated learning. In *International conference on artificial intelligence and statistics* (pp. 2938-2948). PMLR.

[5] Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30.

[6] Liu, Z., Guo, J., Yang, W., Fan, J., Lam, K. Y., & Zhao, J. (2022). Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data*. [Crossref]

[7] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., ... & Roselander, J. (2019). Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1, 374-388.

[8] Mazzocca, C., Romandini, N., Mendula, M., Montanari, R., & Bellavista, P. (2023). TruFLaaS: Trustworthy federated learning as a service. *IEEE Internet of Things Journal, 10*(24), 21266-21281. [Crossref]

[9] Chen, Y., Su, L., & Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems, 1*(2), 1-25. [Crossref]

[10] Singh, P., Singh, M. K., Singh, R., & Singh, N. (2022). Federated learning: Challenges, methods, and future directions. In *Federated Learning for IoT Applications* (pp. 199-214). Cham: Springer International Publishing. [Crossref]

[11] Fusco, P., Rimoli, G. P., & Ficco, M. (2025). TinyML and Federated Learning for Resource-Constrained Medical Devices. In *Artificial Intelligence Techniques for Analysing Sensitive Data in Medical Cyber-Physical Systems: System Protection and Data Analysis* (pp. 113-126). Cham: Springer Nature Switzerland. [Crossref]

[12] Devi, M., Dhaya, R., Kanthavel, R., Algarni, F., & Dixikha, P. (2020). Data Science for Internet of Things (IoT). In *Second international conference on computer networks and communication technologies: ICCNCT 2019* (pp. 60-70). Springer International Publishing. [Crossref]

[13] Dhaya, R., & Kanthavel, R. (2021). Cloud—based multiple importance sampling algorithm with AI based CNN classifier for secure infrastructure. *Automated Software Engineering, 28*(2), 16. [Crossref]

[14] Liu, Y. N., Wang, Y. P., Wang, X. F., Xia, Z., & Xu, J. F. (2019). Privacy-preserving raw data collection without a trusted authority for IoT. *Computer Networks, 148*, 340-348. [Crossref]

[15] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., ... & Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*. [Crossref]

[16] Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017, October). Deep models under the GAN: information leakage from collaborative deep learning. In

Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (pp. 603-618). [Crossref]

[17] Contreras-Castillo, J., Zeadally, S., & Guerrero-Ibánez, J. (2019). Autonomous cars: challenges and opportunities. *IT Professional, 21*(6), 6-13. [Crossref]

[18] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and trends® in machine learning, 14*(1–2), 1-210. [Crossref]

[19] Yin, F., Lin, Z., Kong, Q., Xu, Y., Li, D., Theodoridis, S., & Cui, S. R. (2020). FedLoc: Federated learning framework for data-driven cooperative localization and location data processing. *IEEE Open Journal of Signal Processing, 1*, 187-215. [Crossref]

[20] Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*. [Crossref]

[21] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*. [Crossref]

[22] Kanagavelu, R., Li, Z., Samsudin, J., Yang, Y., Yang, F., Goh, R. S. M., ... & Wang, S. (2020, May). Two-phase multi-party computation enabled privacy-preserving federated learning. In *2020 20th IEEE/ACM international symposium on cluster, cloud and internet computing (CCGRID)* (pp. 410-419). IEEE. [Crossref]

[23] Klaine, P. V., Xu, H., Zhang, L., Imran, M., & Zhu, Z. (2023). A privacy-preserving blockchain platform for a data marketplace. *Distributed Ledger Technologies: Research and Practice, 2*(1), 1-16. [Crossref]

[24] Mahlool, D. H., & Abed, M. H. (2022). A comprehensive survey on federated learning: Concept and applications. *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2022*, 539-553. [Crossref]

[25] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems, 2*, 429-450.

[26] Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., & Luo, W. (2019). Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing, 18*(5), 2438-2455. [Crossref]

[27] Qi, Q., & Chen, X. (2022). Robust design of federated learning for edge-intelligent networks. *IEEE Transactions on Communications, 70*(7), 4469-4481. [Crossref]

[28] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). PMLR. [Crossref]

[29] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 23*(3), 1622-1658. [Crossref]

[30] Samarakoon, S., Bennis, M., Saad, W., & Debbah, M. (2018, December). Federated learning for ultra-reliable low-latency V2V communications. In *2018 IEEE global communications conference (GLOBECOM)* (pp. 1-7). IEEE. [Crossref]

[31] Sattler, F., Wiedemann, S., Müller, K. R., & Samek, W. (2019). Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems, 31*(9), 3400-3413. [Crossref]

[32] Shejwalkar, V., & Houmansadr, A. (2021, January). Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*. [Crossref]

[33] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal, 3*(5), 637-646. [Crossref]

[34] Shokri, R., & Shmatikov, V. (2015, October). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1310-1321). [Crossref]

[35] Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z. (2019). On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

[36] Xie, C., Huang, K., Chen, P. Y., & Li, B. (2019, September). Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

[37] El Ouadrhiri, A., & Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. *IEEE access, 10*, 22359-22380. [Crossref]

[38] Wang, H., Kaplan, Z., Niu, D., & Li, B. (2020, July). Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on computer communications* (pp. 1698-1707). IEEE. [Crossref]

[39] Zhang, F., Liu, X., & Zhang, X. (2019). Communication-efficient federated learning: A comprehensive survey. *IEEE Transactions on Mobile Computing*, 18, 1286-1299. [Crossref]

[40] Short, A. R., Leligou, H. C., & Theocharis, E. (2021, January). Execution of a Federated Learning process within a smart contract. In *2021 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-4). IEEE. [Crossref]

[41] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019, November).

A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security* (pp. 1-11). [Crossref]

[42] Zhang, J., Chen, J., Wu, D., Chen, B., & Yu, S. (2019, August). Poisoning attack in federated learning using generative adversarial nets. In *2019 18th IEEE international conference on trust, security and privacy in computing and communications/13th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)* (pp. 374-380). IEEE. [Crossref]

Prof. Dr. Kanthavel Radhakrishnan is having more than 25 years of academic, research and administrative experience in Asia, Middle East, and Oceania. Currently he is working as Professor of Computer Engineering, School of Electrical and Communications Engineering, PNG University of Technology (Public University & Engineers Australia Accredited) , Lae, Papua New Guinea. He received his Post-Doctoral Fellowship from University of Louisiana, USA, and Ph.D Degree from Anna University, India. He has published more than 20 books, 200 articles in international/national journals/conferences and 05 patents. He is the series editor of a number of book series and serves in various editorial capacities of several international journals. He delivered Key Note addresses in various International Conferences in abroad. His research interests focus on Cloud computing, Machine-deep learning and Intelligent IoT and Wireless Sensor Networks. (Email: kanthavel2005@gmail.com)

Dr. Dhaya Ramakrishnan is having more than 19 years of academic, research and administrative experience in Asia, Middle East, and Oceania. Currently she is working as Senior Faculty of Computer Engineering, School of Electrical and Communications Engineering, PNG University of Technology (Public University & Engineers Australia Accredited), Lae, Papua New Guinea. He received his Post-Doctoral Fellowship from University of Louisiana, USA, and Ph.D Degree from Manonmaniam sundaranar University, India. She has published more than 20 books, 150 articles in international/national journals/conferences and 04 patents. She is the series editor of a number of book series and serves in various editorial capacities of several international journals. Her research interests focus on Cloud computing, Artificial Intelligence, Embedded Systems, Machine -deep learning, Wireless Sensor Networks and Network Security. She received young engineer award by Institution of Engineers (IEI), Kolkata, India. (Email: dhayavel2005@gmail.com)

Dr. R. Adline Freeda has 25 years of teaching and research experience and she has received the M.E in Computer Science and engineering and Ph.D. in Computer science and engineering from Hindustan Institute of Technology & Science, Tamil Nadu, India. Her current research interests include Machine learning, Software Engineering and Cloud Computing. She has published more articles in international/national journals/conferences and patents. (Email: adlinefreeda@gmail.com)