Exploring the Challenges and Opportunities of Hierarchical Federated Learning in Sensor Applications

Melanie Po-Leen Ooi, Annika Hinze, Shaan Nagra
School of Engineering,
University of Waikato,
New Zealand

Shaleeza Sohail
School of Information and Physical Sciences
University of Newcastle,
Australia

Victoria Guiying Huang
National Institute of Water and Atmospheric Research (NIWA),
New Zealand

Nathaniel Hudson, Matt Baughman, Ryan Chard, Kyle Chard, Ian Foster
Department of Computer Science,
University of Chicago,
Argonne National Laboratory
United States

Omer Rana, Theodoros Spyridopoulos
School of Computer Science and Informatics,
Cardiff University,
United Kingdom

Sensor-based monitoring has become ubiquitous in modern society, supporting a number of applications in environmental sciences, urban/city sensing and digital agriculture. AI-based techniques (e.g., machine learning) are effective at extracting actionable information from data generated through such sensors. An example is an automated water irrigation system that uses AI-based techniques on soil quality data to decide how to best distribute water. Unfortunately, these AI-based techniques are costly in terms of hardware resources, and Internet-of-Things (IoT) sensors are resource-constrained with respect to processing power, energy and storage capacity. These limitations can compromise the security, performance and reliability of sensor-driven applications. To address these concerns, cloud computing services can be used by sensor applications for data storage and processing. However cloud-hosted sensor applications that require real-time processing, such as medical applications (e.g., fall detection and stroke prediction), are vulnerable to issues such as network latency due to the sparse and unreliable networks between the sensor nodes and the cloud server [1]. As users approach the edge of a communication network, latency issues become more severe and frequent. A promising alternative is edge computing, which provides cloud-like capabilities at the edge of the network by pushing storage and processing capabilities from centralized nodes to edge devices that are closer to where the data is gathered, resulting in reduced network delays [2, 3].

The most common machine learning approach used in cloud-based applications is Centralized Learning (as shown in Fig. 1(a)) where datasets from different clients are sent to a central cloud for storage and to train a machine learning model. A model trained in this centralized manner is potentially the most accurate model as it has been trained on all of the dataset(s). However, centralized training introduces challenges associated with transferring data to the cloud, such as data privacy and communication overheads.

Conversely, local learning is an alternative where machine learning models are developed directly on devices where the data are hosted, using their own local computing resources (Fig. 1(b)). Since data are not shared among clients or with the server, local learning overcomes privacy and communication overheads of centralized learning. However, local learning alone can struggle with machine learning bias. This is an emergent problem since data at these devices are often non-independent and identically distributed (iid). For instance, many data patterns sensors collect are geographically-related. Thus, sensors from different locations will likely collect different types of data. Learning independently can result in models that do not generalize well which can afflict the knowledge extraction across the entire system.

A third alternative is to apply the Federated Learning (FL) paradigm for distributed learning (Fig. 1(c)). FL is a distributed machine learning technique that enables multiple clients (e.g., mobile devices, IoT devices) to collaboratively train a shared global model without needing their raw data transmitted to the cloud. Instead, local models are trained on each client using their own data, similar to local learning. Where FL diverges from local learning is that the server will periodically collect model updates from client devices and aggregate them to update the global model, which is then redistributed to client devices for further training. FL has shown to work well in the face of non-iid data distributions which are common in sensor applications. Therefore, FL is a promising solution for several challenges associated with decentralized machine learning, such as data privacy, communication efficiency, scalability, data unavailability or heterogeneity, system heterogeneity in terms of computation power, storage, and energy availability, computation efficiency and better model performance [4, 5].

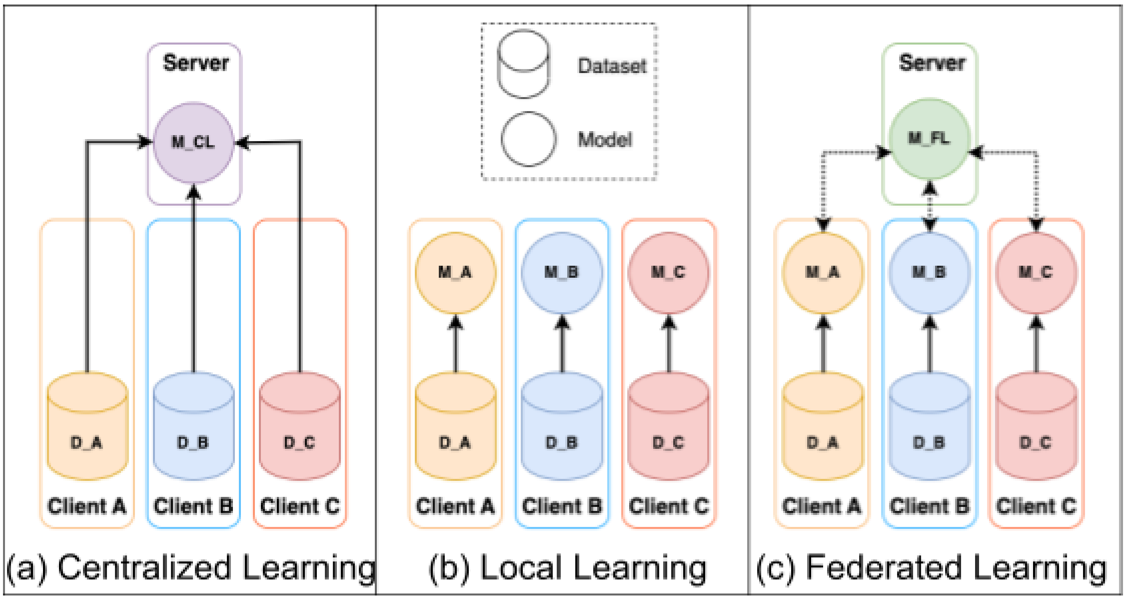

Fig. 1. Comparison of Centralized, Local, and Federated Learning Architectures. Federated learning enables collaborative learning among multiple edge devices without compromising data privacy. Centralized learning relies on a central server to train a global model on data from all devices, while local learning trains multiple local models on disjoint subsets of data

Standard FL (Fig. 2(a)) involves sharing model updates with a central server, which can result in high communication costs, especially if the communication quality (i.e., low data transfer rate, unreliable connection) between sensors and the central server is poor. This can cause problems for individual sensors and application performance, especially if the sensor application has power constraints and limited connectivity [6]. One possible solution is to expand the infrastructure and use hierarchical FL. Under hierarchical FL, there are additional aggregation nodes that may have stronger connections to nearby sensors. These aggregation nodes also participate in additional levels of aggregation to make sure knowledge is learned throughout the entire system. Hierarchical FL overcomes some of these issues by allowing data to remain on edge devices, such as sensors and user owned devices. It enables optimization of computational and communication overheads that can be customized to suit the needs of specific applications and networks.

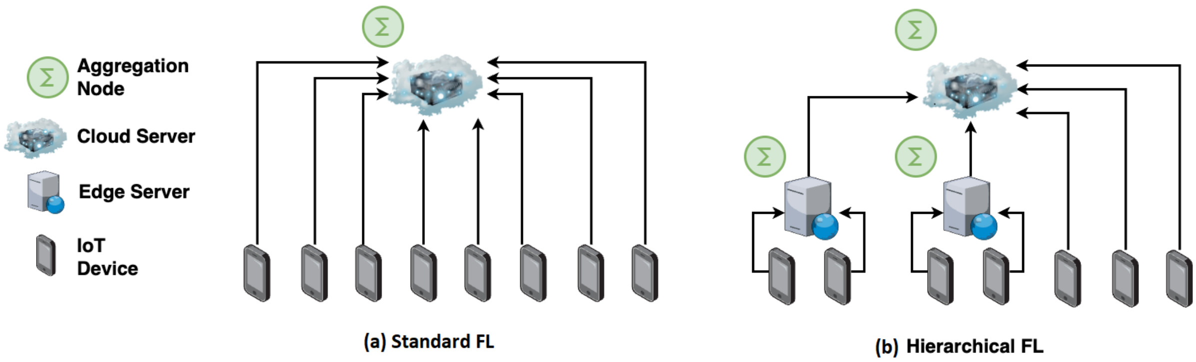

Fig. 2. Illustration of (a) standard FL and (b) hierarchical federated learning; and their interaction between IoT devices, edge and cloud servers. The aggregation nodes are where the local model updates from the participating devices or intermediary nodes are combined to form a global model

Different approaches of hierarchical FL can be employed by making use of available networking hubs, such as a cloud server that is accessed over the internet and edge servers, which are computer servers located closer to devices that it serves at the network edge (Fig. 2(b)). One option is to aggregate models within a cloud server, while other options can involve hierarchical aggregation at the edge server which sends model updates to the cloud for further aggregation or storage. This allows local processing power and storage capacity to perform distributed learning without relying solely on a central server, resulting in improved scalability, reduced latency and privacy preservation of sensitive data.

While hierarchical FL has shown potential in various domains, including precision agriculture and environmental monitoring, [4] points out that very few production FL applications have been reported, with most work being proof-of-concept prototypes. The paradigm itself is relatively new and its implementation requires careful consideration of various issues such as data inequality [7 - 8] and malicious attacks [9 - 11]. Another important challenge to overcome with FL is the cost of adoption and the incentives to incorporate FL into existing systems. In FL, devices must contribute their computational resources and data to train a model. However, some organizations may be reluctant to participate due to the computational resources needed to participate and data transfer and communication costs. Furthermore, there may not be the right incentives to join if a benefit from the global model's performance is not properly recognized. For example, an industry leader may be less incentivized to adopt multi-organizational FL if they are already the primary source of data as their marginal benefit would be significantly lower.

Various methods have been proposed to address these challenges to incentivize participation in FL. These methods include providing monetary or non-monetary rewards for contributing data, ensuring that the global model's performance benefits all participants, and providing greater control over the use of data by participants. For example, [12] proposed a reward-based mechanism that provides tokens to devices that contribute data to the FL model, which can then be used to purchase services or products. [13] propose a privacy-preserving FL model that allows participants to retain control over their data and provides incentives for contributing high-quality data.

In this article, we present a case study of the feasibility and benefits of FL for precision agricultural spraying and extend it to a larger conceptual hierarchical FL architecture for smart farming. We also discuss ongoing efforts to improve hierarchical FL systems using standardized quality frameworks to systematically identify and prioritize the efforts. Finally, we cover open problems associated with this technology and highlight its potential to transform numerous domains such as precision agriculture, environmental monitoring, healthcare, and smart infrastructure. Our goal is to provide a roadmap for further development of hierarchical FL in instrumentation and measurement.

Federated Learning for Precision Spraying

In the previous section, we discussed the different types of learning architectures that can be applied to edge devices. In this section, we present a case study involving precision spraying in agriculture (see Fig. 3) to highlight the benefits of FL.

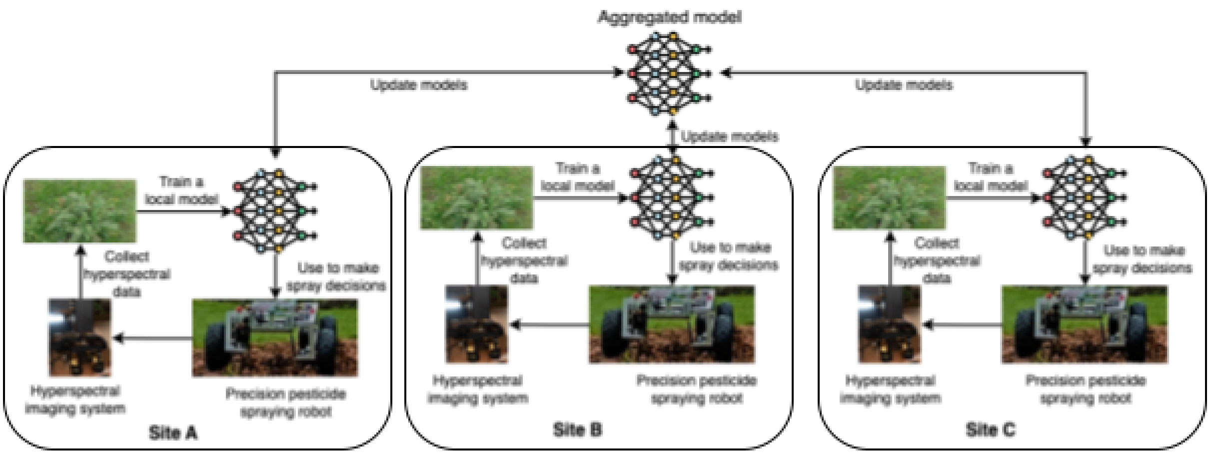

Fig. 3. Precision spraying prototype using edge federated learning.

Traditional spraying methods often apply pesticides uniformly across an entire field, leading to waste and potential harm to crops. In contrast, precision spraying uses cameras, optical sensors, and GPS receivers to provide accurate location data, enabling robots to apply spray only where necessary. However, the advanced data processing and machine learning required for precision spraying can be resource-intensive, making it challenging to perform all operations on the sensor or robot alone.

Data collected from sensors can be processed using cloud computing relying on connectivity, but this can be a challenge in remote environments. By relying on computational resources at the edge server, data can be processed in real-time or limited-time to accelerate the decision-making process. FL can further improve spray precision by enabling learning from different data sources (e.g., across different locations with different distributions of plant/weed species) while preserving data privacy and security. This is particularly important for farmers who are concerned about data privacy and potential cyberattacks. Furthermore, FL can be flexibly performed on a network of edge devices potentially of different types, which can scale up or down to meet the needs of different agricultural operations, making it a valuable tool for precision spraying.

In this feasibility study, we limit the discussion to a single source of information, which is a near-infrared hyperspectral imaging system that is used to guide the operations of a mobile precision pesticide spraying robot at each of the three pasture sites (Site A, B and C). Each robot processes the input information using a local machine learning model (e.g., an image classifier implemented by a Convolutional Neural Network). The intent is to have the robot apply spray to a plant only if the plant is classified as a weed, ensuring minimal impact on crops. Storing a machine learning model locally ensures that spray operations can continue in areas without network coverage. Although the local model is trained using local data (~71 Megabytes), it periodically receives updates (~0.04 Megabytes) from the server via FL to incorporate knowledge learned from other pasture sites. With FL, communication is significantly reduced from 71 Megabytes to 0.04 Megabytes. Different forms of computation resources reside in each pasture site, with the model potentially being trained on the robot using single-board computational devices such as Raspberry Pi or Jetson Nano. The locally-trained models from each pasture site are then aggregated using Federated Averaging [14], a popular aggregation algorithm in FL. The pasture sites are interconnected using communication infrastructure for sharing information and enabling FL.

The dataset used during our evaluation consisted of four labelled classes: three species of pastoral weed and a background class of grass (see Table 1). The evaluation results for our case study with different machine learning methods (discussed in the previous section and Fig. 1) are shown in Fig. 4, whereby FL achieves 96% accuracy, a result that is comparable to centralized learning where all the data are present, and a marked improvement compared to local learning where only local data are present. Our result confirms that FL is capable of addressing the dilemma between network latency, bandwidth limitations, data privacy and data sharing while providing comparable model performance.

Table 1. Pasture Image Dataset with Imbalanced Class Distributions and Disparate Volumes of Data Across Sites

Pasture Site	A	B	C
Number of labelled samples	60072	30240	6232
Number of labelled classes (W: pastoral weeds, G: background grass)	4 (3W+G)	4 (3W+G)	2 (1W+G)

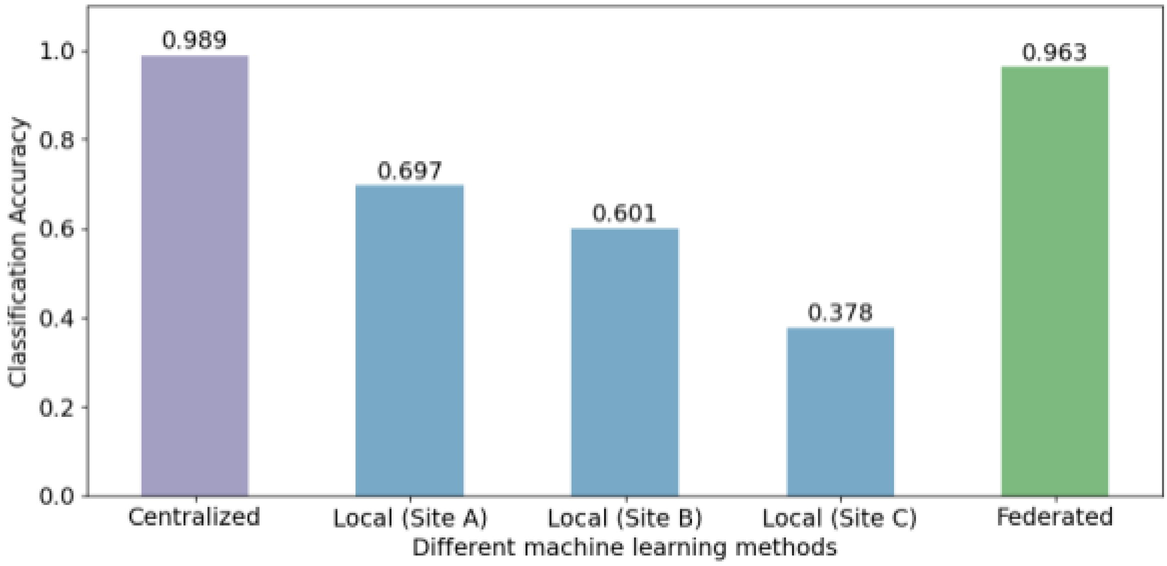

Fig. 4. Classification accuracy comparison between centralized, local, and federated machine learning approaches trained on dataset described in Table 1.

A Hierarchical Federated Learning Architecture for Smart Farming

We now consider broadening the precision spraying use case to a larger conceptual hierarchical FL architecture for smart farming. Precision agriculture often involves gathering heterogeneous data from various sensors within a farm. By leveraging FL, farmers can benefit from insights gained from other farms without compromising the privacy of their own raw data. This will enable them to enhance crop yields and minimize wastage.

Fig. 5 illustrates the physical, cyber and networked (interconnected) view of a precision agriculture robot at work. Fig. 5(a) shows the "physical world" where the robot has sensors on board and navigates to perform tasks such as harvesting, weeding, and spraying. Fig. 5(b) represents the "cyber world" where the robot processes input information from its sensors using a local machine learning model to perform high priority tasks that must continue even when the communication fails (i.e. navigation). The robot also monitors the quality of service for communication and decides whether to pass on further information or receive information from the cloud. Fig. 5(c) is the "interconnected world" where a group of robots is connected via a communication infrastructure for FL to share knowledge and improve their respective local machine learning models based on collective learned experiences.

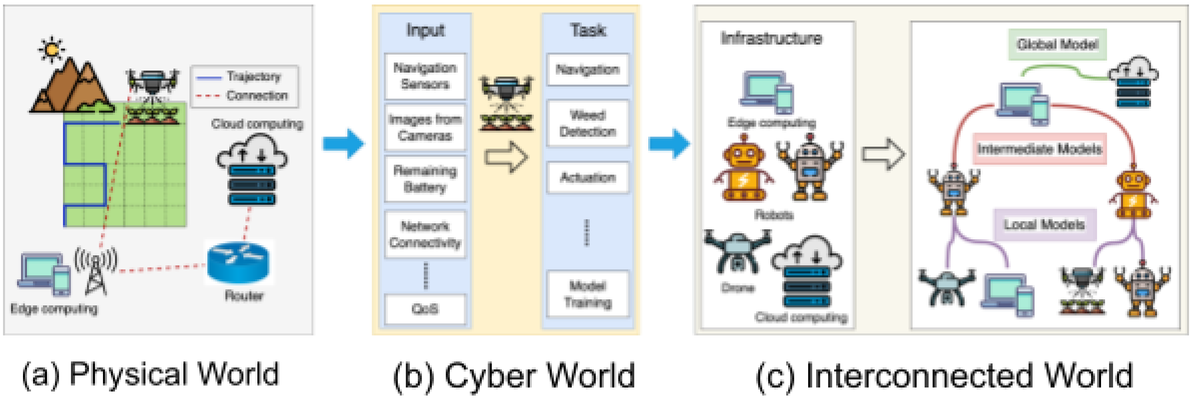

Fig. 5. Illustration of hierarchical federated learning in precision agriculture.

The types of sensor systems used for smart farming are heterogeneous. For example, a smart farm may rely heavily on IoT sensors and cameras to collect and process data related to soil quality and plant visual information. Additionally, smart farms may use multiple data sources to optimize farming operations. These data sources may include historical data, real-time sensor data, and publicly available data to predict soil water levels and future weather patterns with machine learning models. Fig. 6 illustrates an example of a hierarchical FL architecture for smart farming to train lightweight machine learning models on edge devices.

In this example, the architecture uses Apache Kafka, a popular distributed event streaming platform. Kafka is used to collect data from IoT devices and sensors located at the edge of the network, enabling real-time processing and response to changing conditions. Instead of waiting for large amounts of data to be collected and sent to a remote, central server, data streaming enables the sensor devices to transmit the local models to the edge servers in real-time—thus reducing latency and facilitating more efficient use of available network bandwidth. Hierarchical FL allows for adaptive clustering of the edge devices involved in the FL process, whereby groups of devices can be divided into smaller clusters for the training depending on device availability, scalability and power requirements. By making use of Kafka clusters to facilitate communication between the edge devices and the FL framework, fault tolerance can be incorporated with multiple brokers running on different machines to ensure high availability and data replication.

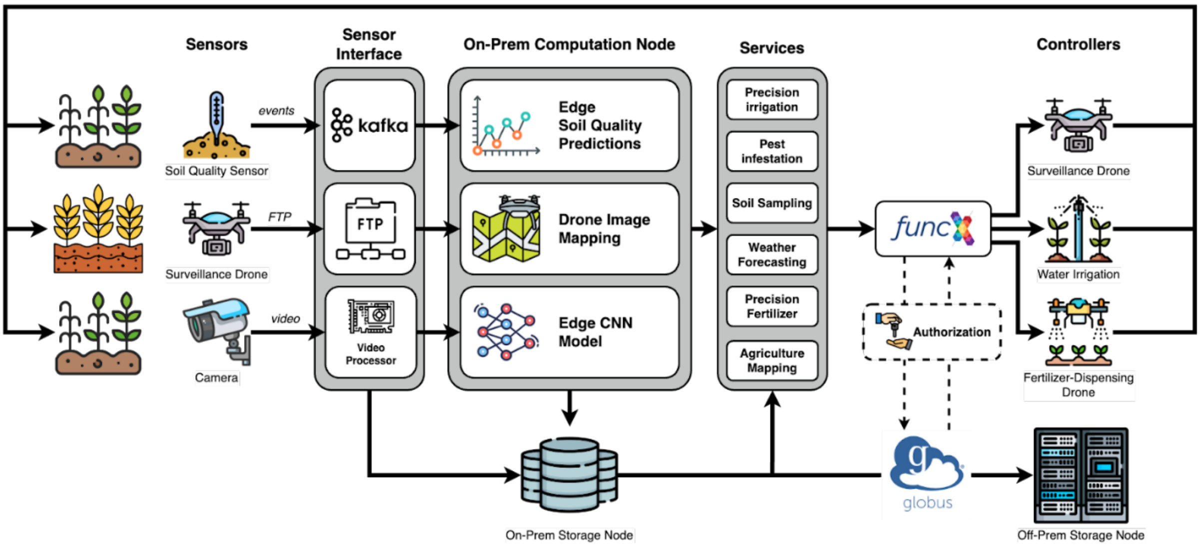

Fig. 6. Illustration of a hierarchical federated learning architecture for smart farming

FuncX [15] is a federated function-as-a-service platform that enables computation, represented as programming functions, to be dispatched for execution on edge resources. FuncX is used here to manage the deployment of the machine learning models to the edge devices and to invoke the models for inference and training. FuncX uses the Parsl [16] parallel programming library to manage the parallel execution of the federated learning tasks on edge devices. It is worth noting that this simple architecture can be easily extended to include heterogeneous resources such as Raspberry Pis and Nvidia Jetsons with GPUs, which can be deployed near the sensors for more local data processing. This introduces interesting research questions on the implicit trade-offs related to such system heterogeneity.

Overall, using a hierarchical FL architecture for edge devices has the potential to revolutionize smart farming by enabling more efficient and effective data sharing, processing and decision-making, resulting in improved crop yields and reduced water waste. It is also resilient against issues such as unreliable network connectivity, which is an important consideration as limited network coverage is commonly encountered in agricultural land.

Similar architectures can be developed for other applications such as:

- **Environmental monitoring.** Hierarchical FL could enable better analysis of data collected from remote sensors and devices, leading to better understanding and management of natural resources.
- **Healthcare**. Medical sites can collaborate and share machine learning models to improve the accuracy of diagnosis and treatment without having to share sensitive patient information.
- **Wearable monitoring devices**. IoT devices are integrated into devices to monitor human biometric data such as heart rate, temperature and movement patterns aiming to identify potential safety hazards and improve worker safety in the hazardous work environment.

Application Requirements for Hierarchical Federated Learning

Hierarchical FL systems have great potential to realize new sensor-driven applications, as discussed in the previous section. However, the technology faces numerous challenges due to the diverse nature of these applications, and the equally diverse performance requirements. Some of these requirements can conflict with one another. To address the needs of these systems systematically, we recommend using two widely recognized standards for software quality and data quality, which are the ISO/IEC 25012 [17] and ISO/IEC 25010 [18] frameworks respectively. By applying these frameworks, we can identify and prioritize the requirements, design appropriate performance metrics, and develop techniques and algorithms that optimize the trade-offs between conflicting requirements. This will help ensure that hierarchical FL systems are reliable, efficient, and effective, and can be deployed in a wide range of real-world applications.

- **Functionality** is a crucial metric that measures the system's ability to perform consistently and accurately over time. This metric is closely related to data quality since poor-quality data can result in unreliable and inaccurate machine learning models. Optimizing hierarchical FL parameters, applying model personalization and transfer learning techniques, and adapting the models to the specific characteristics of each device can help improve functionality.
- **Reliability** is important for edge devices, which often operate under intermittent connectivity, mobility, and resource-constrained conditions [19]. These challenges can make it difficult to develop efficient and reliable FL systems. To overcome these challenges, researchers are exploring techniques such as task offloading [20], adaptive sampling [21], and reinforcement learning to optimize the system performance and adapt to changing edge computing environments [22]. Redundancy can be introduced by configuring the edge devices into different clusters, which may improve reliability in hierarchical FL.
- **Performance efficiency** can be evaluated by quantifying the computation and communication overheads. In hierarchical FL systems, the edge server acts as a coordinator, responsible for aggregating and processing the data from multiple edge devices [23]. This can create a huge computational and communication burden on the edge server, which can result in high latency, reduced system performance, and increased communication costs. To mitigate these challenges, researchers are exploring techniques such as partitioning and scheduling to distribute the computation and communication load across multiple edge servers and among cloud and edge servers [6]. The use of data streaming techniques helps to further reduce communication overheads while also increasing the reliability of the edge systems. Furthermore, due to resource constraints, mobility, and varying edge computing environments, engineers should consider metrics like system scalability, cost/benefit balance, and application size on the edge device to evaluate resource utilization. Techniques like task offloading, adaptive sampling, and reinforcement learning can further optimize system performance and adapt to changing edge computing environments. The convergence rate of the learning process is also a critical consideration.
- **Compatibility** is essential in terms of device heterogeneity, where edge devices may not have the same type of sensors and may run on different operating systems and software. Edge devices used in FL systems may differ in terms of their hardware

capabilities, such as CPU processing power, memory, and battery life, as well as their software configurations, such as operating systems and libraries. This heterogeneity can lead to statistical heterogeneity in the data collected from the devices, making it challenging to develop models that generalize well across all devices [24]. To overcome this challenge, researchers are exploring methods such as transfer learning [25] and model personalization [26] that can adapt the models to the specific characteristics of each device.

- **Security** is a significant concern for hierarchical FL systems. FL provides some level of privacy, but to protect data and models from malicious actors, techniques like secure aggregation, differential privacy, blockchain, and homomorphic encryption can be applied. Hierarchical FL systems may have edge servers from different sources, which makes the system more vulnerable to security threats such as data leakage, model poisoning, and inference attacks [23]. To address these challenges, researchers are exploring techniques such as secure aggregation [27], differential privacy [28], and homomorphic encryption [29] to ensure that the data and models are protected from malicious actors. Metrics like confidentiality, currentness, and reliability can be used to evaluate the severity of security challenges, as defined by the ISO/IEC 25010 standard for data security and accessibility.
- **Dynamic infrastructure** is a common problem in hierarchical FL systems. In these settings, the aggregation nodes are not necessarily static or fixed over time. It may be necessary for the aggregation node to change over time if a node is rendered offline due to low battery or a poor communication channel. This makes ensuring stable hierarchical FL difficult. Strategies to prevent this can be resource-aware real-time decision-making where edge devices coordinate among themselves to decide how communication should be done among them.
- **Software support.** Currently, common FL frameworks (e.g., Flower, PySyft) do not natively support hierarchical FL, especially for highly dynamic systems where infrastructure may change over time.

Opportunities and Open Problems

The opportunities and future research directions for using FL for sensing and measurement can be organized along three key axes: (i) resource management and coordination of devices that make up the FL system; (ii) data management and access; (iii) application-specific considerations, such as the use of UAVs [30] and Internet of Medical Things [23]. In this section, we elaborate on the first two axes, which have general applicability across different domains. The third is domain-specific and thus not included in this general roadmap.

Resource Management & Coordination

FL opens up new methods for supporting **fault tolerance** of IoT devices. The use of multiple IoT devices increases the completeness of data and facilitates the detection and correction of erroneous readings and faults. An FL system capable of supporting multiple heterogeneous devices and able to recover from faults is reported in [31]. A hierarchical FL system able to account for the hierarchy of edge servers may result in multiple memberships for IoT devices requiring recovery and adaptation at several levels.

Another direction is the integration of **self-adaptivity** into IoT devices using hierarchical FL. IoT devices become self-adaptive by learning from their past operations and performance data, and can detect changes in environmental conditions, predict equipment failure and automatically adjust device parameters [32]. As IoT devices collect knowledge about their environment, they must share knowledge to better understand their environment and its dynamics. However, a collective system of self-aware devices does not have a centralized knowledge base. FL could facilitate sharing of knowledge between devices as a collective. Unlike context-aware systems, which typically assume a ground-truth-based environmental context that is true for all IoT devices, it is explicitly acknowledged that self-aware devices in the collective systems will have different experiences of a shared environment [33]. A federated system can be expanded to incorporate information for edge servers to provide self-adaptiveness for different branches of the edge computing infrastructure, and to provide context on how the information was gathered.

Hierarchical FL architectures are inherently resilient against network latency and bandwidth limitations – opening up new opportunities for network optimization due to the heterogeneity of edge environments. For example, network routing needs to be dynamically adapted and optimized to traffic patterns and topologies to improve network throughput and latency. Meanwhile, new network designs and paradigms can be introduced, such as Software-defined Networking and Blockchain.

New algorithms and techniques that are specifically designed to **optimize resource allocation** in edge computing environments can dynamically allocate computational and communication resources to different tasks based on their priority and application-specific requirements. This allows for real-time balancing of multiple objectives. For example, a healthcare application using a large number of medical sensors to collect, adapt and react to medical information in real-time requires security robustness and resource optimization for computational and communication tasks [34]. The energy consumption of IoT devices and edge computing infrastructure can be optimized by dynamically adapting the computational workload and communication overhead to the available energy and power constraints.

Exploring **multi-objective optimization** techniques such as evolutionary algorithms and Pareto optimization is another direction of research that is crucial due to the complexity of hierarchical FL systems. There is a need to balance multiple objectives and constraints, such as communication overhead, energy consumption, privacy preservation, and security while creating a global machine learning model with limited and heterogeneous resources. This requires the coordination of multiple clients and servers at different layers of the system, creating a complex interaction network. Additionally, the underlying technologies used, such as wireless communication, data storage and processing, and machine learning algorithms, introduce their sets of constraints and trade-offs that need to be considered in the optimization process, where not only performance but scalability needs careful consideration when optimizing these conflicting objectives.

Data Management & Access

While FL adds a layer of **privacy** around user data by eliminating the need to share data with others, sharing model updates do not offer privacy guarantees as individual data points can be reconstructed [35]. To address this issue, privacy-preserving FL can be expanded using techniques such as differential privacy, homomorphic encryption and digital signatures

[23][36]. These techniques enable FL in edge computing environments while preserving the privacy of data and participants. For instance, differential privacy provides privacy guarantees for medical image analysis. In unstable edge computing environments, such as smart healthcare, a privacy protection scheme is proposed that provides gradient privacy and resistance to collusion and replay attacks for Internet of Medical Things (IoMT) [23]. A privacy protection technique for a hierarchical FL system needs to be computationally efficient and adaptable to the underlying learning algorithms with an ability to effectively scale to a large number of clients and servers.

Fig. 7 summarises some of the opportunities presented by hierarchical FL that make it an exciting area of research for engineers and academicians in the field of instrumentation and measurement.

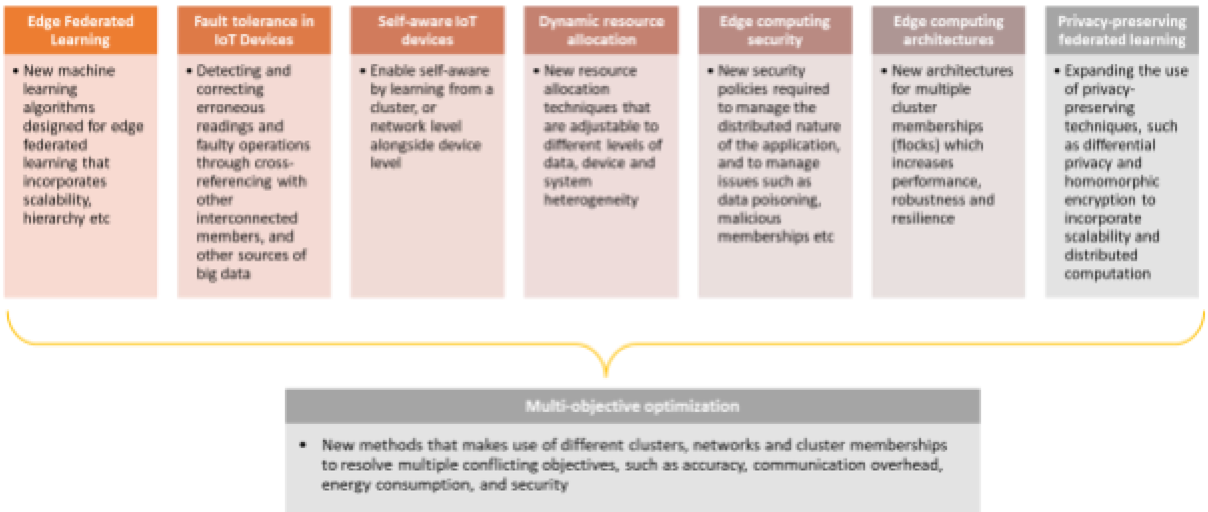

Fig. 7. Future directions for hierarchical federated learning in sensor applications

Conclusion

Sensor-based data collection has continued to increase over recent years, primarily due to the availability of low-cost sensing environments and increasing integration of sensing with Cloud-hosted analytics. However, latency constraints between sensing and analytics can limit benefits, thus requiring the availability of *edge clouds,* capable of undertaking partial analysis in proximity to sensing devices. Such infrastructure provides additional opportunities to deploy machine learning closer to where data is captured, providing initial analysis at lower computational cost and at lower latency. Challenges in realising such a machine learning environment are investigated, along with the benefits and limitations of realising this in practice. Hierarchical federated learning is an emerging technology that enhances the intelligence of sensor systems in many applications such as environmental quality management, personalized healthcare devices, and precision agriculture. Secure, resilient and robust sensor systems that support real-time, data-driven decision-making are valuable infrastructure for helping us reduce carbon emissions (SDG 13 - Climate Action), promote sustainable urbanization (SDG 11 - Sustainable Cities and Communities) and improve healthcare outcomes (SDG 3 - Health and Well-being). We urge researchers and practitioners to collaborate on the opportunities and open problems presented in this article to realize the full potential of this technology.

References

[1] M. Alrazgan, “Internet of medical things and edge computing for improving healthcare in smart cities,” Mathematical Problems in Engineering, vol. 2022, 2022.

[2] S. Hamdan, M. Ayyash, and S. Almajali, “Edge-computing architectures for Internet of Things applications: A survey,” Sensors, vol. 20, no. 22, p. 6441, 2020.

[3] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey, future generation computer systems,” 2019.

[4] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik, and L. Zhu, “A systematic literature review on federated machine learning: From a software engineering perspective,” ACM Computing Surveys (CSUR), vol. 54, no. 5, pp. 1–39, 2021.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings et al., “Advances and open problems in federated learning,” Foundations and Trends® in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.

[6] W. Liu, B. Cao, L. Zhang, M. Peng, and M. Daneshmand, “A distributed game theoretic approach for blockchain-based offloading strategy,” in ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020, pp. 1–6.

[7] L. Corinzia, A. Beuret, and J. M. Buhmann, “Variational federated multitask learning,” arXiv preprint arXiv:1906.06268, 2019.

[8] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, “Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records,” Journal of biomedical informatics, vol. 99, p. 103291, 2019.

[9] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for privacy-preserving machine learning,” in proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175 – 1191.

[10] R. C. Geyer, T. Klein, and M. Nabi, “Differentially private federated learning: A client level perspective,” arXiv preprint arXiv:1712.07557, 2017.

[11] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, “Efficient and privacy-enhanced federated learning for industrial artificial intelligence,” IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6532– 6542, 2019.

[12] I. Martinez, S. Francis, and A. S. Hafid, “Record and reward federated learning contributions with blockchain,” in 2019 International Conference on Cyber-enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2019, pp. 50–57.

[13] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, “A learning-based incentive mechanism for federated learning,” IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6360–6368, 2020.

[14] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, “A performance evaluation of federated learning algorithms,” in Proceedings of the second workshop on distributed infrastructures for deep learning, 2018, pp. 1–8.

[15] R. Chard, Y. Babuji, Z. Li, T. Skluzacek, A. Woodard, B. Blaiszik, I. Foster, and K. Chard, “Funcx: A federated function serving fabric for science,” in Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, 2020, pp. 65–76.

[16] Y. Babuji, A. Woodard, Z. Li, D. S. Katz, B. Clifford, R. Kumar, L. Lacinski, R. Chard, J. M. Wozniak, I. Foster et al., “Parsl: Pervasive parallel programming in python,” in Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing, 2019, pp. 25–36.

[17] “ISO/IEC 25012,” https://iso25000.com/index.php/en/iso-25000-standards/iso-25012, Accessed: 2023-14-04.

[18] “ISO/IEC 25010,” https://iso25000.com/index.php/en/iso-25000-standards/iso-25010, Accessed: 2023-14-04.

[19] X. Zhang, X. Zhu, J. Wang, H. Yan, H. Chen, and W. Bao, “Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks,” Information Sciences, vol. 540, pp. 242–262, 2020.

[20] P. Tam, S. Math, and S. Kim, “Optimized multi-service tasks offloading for federated learning in edge virtualization,” IEEE Transactions on Network Science and Engineering, vol. 9, no. 6, pp. 4363–4378, 2022.

[21] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, “Tackling system and statistical heterogeneity for federated learning with adaptive client sampling,” in IEEE INFOCOM 2022-IEEE conference on computer communications. IEEE, 2022, pp. 1739–1748.

[22] W. Yang, W. Xiang, Y. Yang, and P. Cheng, “Optimizing federated learning with deep reinforcement learning for digital twin empowered industrial iot,” IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1884–1893, 2022.

[23] R. Wang, J. Lai, Z. Zhang, X. Li, P. Vijayakumar, and M. Karuppiah, “Privacy-preserving federated learning for Internet of Medical Things under edge computing,” IEEE Journal of Biomedical and Health Informatics, 2022.

[24] C. Xu, Y. Qu, Y. Xiang, and L. Gao, “Asynchronous federated learning on heterogeneous devices: A survey,” arXiv preprint arXiv:2109.04269, 2021.

[25] W. Zhang and X. Li, “Data privacy preserving federated transfer learning in machinery fault diagnostics using prior distributions,” Structural Health Monitoring, vol. 21, no. 4, pp. 1329–1344, 2022.

[26] J. A. Ruiz-Mill´an, E. Mart´ınez-C´amara, M. Victoria Luz´on, and F. Herrera, “Personalised federated learning with bert fine tuning. case study on twitter sentiment analysis,” in Advances in Deep Learning, Artificial Intelligence and Robotics: Proceedings of the 2nd International Conference on Deep Learning, Artificial Intelligence and Robotics,(ICDLAIR) 2020. Springer, 2022, pp. 193–202.

[27] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E Ali, B. Guler, and S. Avestimehr, “Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning,” Proceedings of Machine Learning and Systems, vol. 4, pp. 694–720, 2022.

[28] W. Xue, Y. Shen, C. Luo, W. Xu, W. Hu, and A. Seneviratne, “A differential privacy-based classification system for edge computing in iot,” Computer Communications, vol. 182, pp. 117–128, 2022.

[29] J. Park and H. Lim, “Privacy-preserving federated learning using homomorphic encryption,” Applied Sciences, vol. 12, no. 2, p. 734, 2022.

[30] J. Tursunboev, Y.-S. Kang, S.-B. Huh, D.-W. Lim, J.-M. Kang, and H. Jung, “Hierarchical federated learning for edge-aided unmanned aerial vehicle networks,” Applied Sciences, vol. 12, no. 2, p. 670, 2022.

[31] J. ´A. Morell and E. Alba, “Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices,” Future Generation Computer Systems, vol. 133, pp. 53–67, 2022.

[32] S. Attarha and A. F¨orster, “Service management for enabling self-awareness in low-power iot edge devices,” in 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). IEEE, 2022, pp.146–147.

[33] P. Lewis, K. L. Bellman, C. Landauer, L. Esterle, K. Glette, A. Diaconescu, and H. Giese, "Towards a framework for the levels and aspects of self-aware computing systems," Self-Aware Computing Systems, pp. 51–85, 2017.

[34] L. G. F. da Silva, D. F. Sadok, and P. T. Endo, "Resource optimizing federated learning for use with iot: a systematic review," Journal of Parallel and Distributed Computing, 2023.

[35] J. Cui, H. Zhu, H. Deng, Z. Chen, and D. Liu, "Fearh: Federated machine learning with anonymous random hybridization on electronic medical records," Journal of Biomedical Informatics, vol. 117, p. 103735, 2021.

[36] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," Scientific reports, vol. 12, no. 1, p. 1953, 2022.