Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version 11 January, 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.011100

Federated Learning for Human Activity Recognition: Overview, Advances, and Challenges

Ons Aouedi[1], Alessio Sacco[2], Latif U. Khan[3], Dinh C. Nguyen[4], and Mohsen Guizani[3]
(Fellow, IEEE)

[1] SnT, University of Luxembourg, Luxemburg (e-mail: ons.aouedi@uni.lu)
[2] DAUIN, Politecnico di Torino, Turin, Italy (e-mail: alessio_sacco@polito.it)
[3] Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, United Arab Emirates, (e-mail: latif.u.khan2@gmail.com, mguizani@ieee.org)
[4] Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, USA (e-mail: dinh.nguyen@uah.edu)

CORRESPONDING AUTHOR: Alessio Sacco (e-mail: alessio_sacco@polito.it).

ABSTRACT Human Activity Recognition (HAR) has seen remarkable advances in recent years, driven by the widespread use of wearable devices and the increasing demand for personalized healthcare and activity tracking. Federated Learning (FL) is a promising paradigm for HAR that enables the collaborative training of machine learning models on decentralized devices while preserving data privacy. It improves not only data privacy but also training efficiency as it utilizes the computing power and data of potentially millions of smart devices for parallel training. In addition, it helps end-user devices avoid sending users' private data to the cloud, eliminates the need for a network connection, and saves the latency of back-and-forth communication. FL also offers significant advantages for communication by reducing the amount of data transmitted over the network, alleviating network congestion and reducing communication costs. By distributing the training process across devices, FL minimizes the need for centralized data storage and processing, leading to more scalable and resilient systems. This paper provides a comprehensive survey of the integration of FL into HAR applications. Unlike existing reviews, this paper uniquely focuses on the intersection of FL and HAR, providing an in-depth analysis of recent advances and their practical implications. We explore key advances in FL-based HAR methodologies, including model architectures, optimization techniques, and different applications. Furthermore, we highlight the major challenges and future research questions in this domain, such as model personalization and robustness, privacy concerns, concept drift, and the limited capacity of edge devices.

INDEX TERMS Federated Learning, Machine Learning, Human Activity Recognition, Data Privacy

I. INTRODUCTION

The spread of wearable technology and Internet of Medicine (IoMT) sensors is constantly creating a massive amount of health-related data [1]. These data have the potential to completely change personalized health monitoring and intervention. However, there are many challenges to overcome before these data can be managed and used in a centralized way, especially when it comes to privacy, latency, and scalability. Therefore, Human Activity Recognition (HAR) coupled with Federated Learning (FL) is a significant advance and holds great promise for our daily lives [2, 3]. It is essential to model user behavior in many different applications, including fitness tracking, fall detection, and ubiquitous health monitoring. HAR is a key component of smart healthcare applications and involves classifying and predicting human activities from sensor data. This finds applicability in the monitoring, rehabilitation, and early detection of abnormalities in patients [4]. Sensitive personal data processing is often needed for these applications, which poses serious privacy issues. FL has been used to address this problem by enabling several devices/clients to learn a shared model cooperatively without sharing private data [5]. Instead of transferring raw, sensitive data to a central location, FL operates by training machine learning (ML)/Deep Learning (DL) models on each device, and then sharing and aggregating the model updates. As shown in Figure 1, FL-based approaches significantly reduce the risk of sensitive data exposure and mitigate the need for extensive data transmission, addressing privacy and latency issues. The figure also shows that centralized learning involves collecting all data in a single location for training, which, while enabling fast convergence and centralized control, poses significant privacy risks and requires extensive communication resources. In contrast, FL allows for on-device training, requiring only the transfer of learning updates. This approach improves privacy by keeping data local, reduces communication needs, and is better suited to the decentralized nature of B5G and 6G networks, despite slower convergence and challenges related to system heterogeneity. Technologies such as 5G, 6G, and visible light communication schemes are integral to efficient training and data transfer in both centralized and FL approaches. The trained HAR models serve end users or systems and can be securely stored in a blockchain for future use.

This introduction of FL to HAR applications opens new avenues for providing more personalized and effective healthcare solutions [6]. Its decentralized structure makes it a great fit for future smart healthcare systems, since it facilitates a more effective, scalable, and private method of learning from health-related data. Furthermore, the combination of FL and HAR helps to offer intelligent solutions anywhere and on any scale. This is reflected in the increasing number of research papers dedicated to exploring FL's potential in this domain [7]. Because of these qualities, FL can provide several advantages for HAR applications, as shown in Figure 2, FL can significantly improve HAR systems by facilitating real-time processing, enhancing privacy preservation, and leveraging the computational power of edge devices. These advantages make FL a compelling solution for advancing HAR technologies, leading to more accurate and efficient activity recognition systems that are well-suited for deployment in real-world scenarios.

A. Existing Surveys and Tutorials

The article in [8] explores the concept of FL in detail, focusing on critical system components such as data distribution, the ML model, privacy mechanisms, and communication architecture. This comprehensive survey offers a foundational understanding of FL and its various facets. Similarly, the work presented in [9] takes an architectural perspective on the FL concept, analyzing its basic applications in business contexts. This survey highlights how FL can be integrated into business models and the potential benefits it can bring. Further studies such as [10] [11] examine the structure of FL, its software, platforms, and protocols, shedding light on the potential challenges that may arise during FL deployments. These papers are crucial as they provide insights into the technical infrastructure required for effective FL deployment and identify gaps that need to be addressed. Although it provides a comprehensive view of FL software engineering practices, it does not delve into the specific challenges and solutions related to HAR using FL. In addition, the use of FL within mobile edge networks was examined in [12], with an emphasis placed on addressing challenges in the FL application and understanding FL's contributions to optimizing edge networks. This study is significant in highlighting how FL can be used to improve network performance. Although it touches on the importance of FL in edge computing, it lacks a detailed discussion on its application in HAR.

Meanwhile, the work in [7] [13] presents a survey on FL within the Internet of Things (IoT) and describes the technical issues in FL designs, as well as the main application of FL in the IoT. These surveys underscore the importance of FL in managing the vast amounts of data generated by IoT devices while maintaining privacy and efficiency. Although HAR is mentioned as a potential application, the discussion is brief and lacks specific examples or detailed analysis. The study conducted in [14] delves into how FL can offer a solution for the future of digital healthcare and, at the same time, highlights the main challenges in this domain. [15] offers a review of the main structures of FL models and only briefly introduces the application of FL in the field of health informatics. In the same direction, the authors in [1] and [16] provide an extensive survey on FL in the context of IoMT. These surveys provide a detailed overview of how FL can be applied to medical devices and systems, enhancing data sharing and analysis while protecting patient privacy. Another study in [14] considers technical issues and prerequisites for the employment of FL in the future landscape of digital health. Moreover, the potential of FL to leverage electronic health records (EHR) data for healthcare applications was proposed in [17]. This study is critical because it explores how FL can be used to improve healthcare outcomes by allowing collaborative analysis of health records without compromising patient privacy. On the other hand, this survey [18] focused on transfer learning methods in the application domains of HAR, where FL-based solutions are partially covered. We summarize the related work and compare it with our paper in Table 1.

Although there has been considerable research effort, to the best of our knowledge, there is a noticeable gap in comprehensive surveys focusing on the use of FL in the HAR domain. Furthermore, the existing literature lacks a holistic taxonomy and a more practical demonstration of the use of FL in evolving HAR systems. These gaps motivate us to conduct a comprehensive investigation of FL integration in the HAR realm. Thus, this paper presents a comprehensive survey on the integration of FL into HAR applications. Unlike existing reviews, this paper uniquely focuses on the intersection of FL and HAR, providing an in-depth analysis of the recent advancements and their practical implications. First, we highlight the key motivations and requirements for the use of FL in HAR. Then we present the design aspects, the architecture, and the FL frameworks. We also furnish an up-to-date survey of the burgeoning applications of FL in HAR. Moreover, we summarize the lessons learned from the survey to provide the reader with deeper insights into the practical application of FL in HAR.

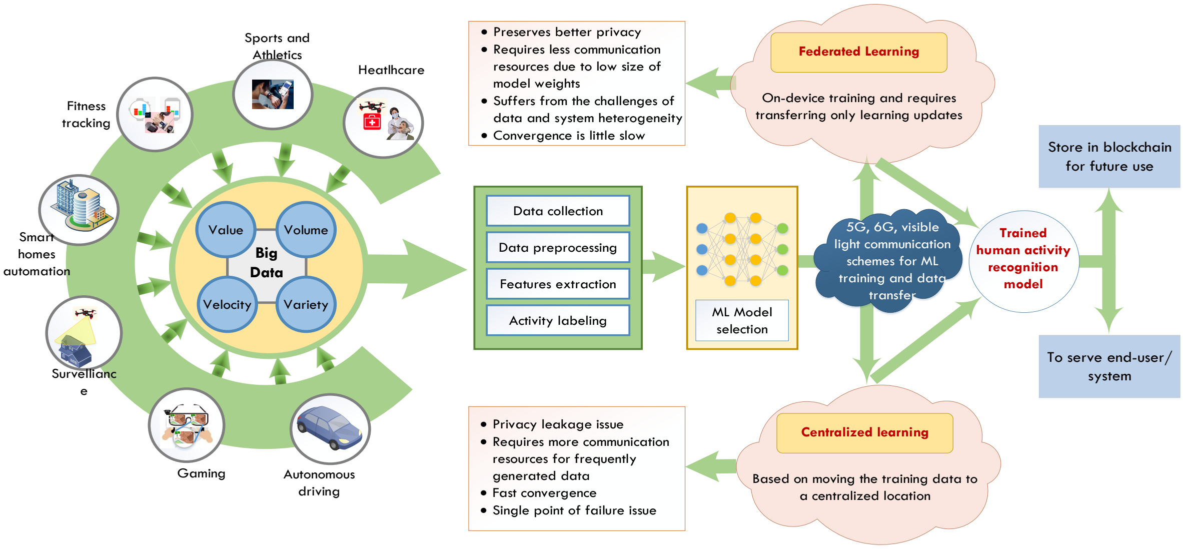
Sports and Athletics
Heatlhcare
Fitness tracking
Smart homes automation
Survelliance
Gaming
Autonomous driving
Value
Volume
Big Data
Velocity
Variety
Data collection
Data preprocessing
Features extraction
Activity labeling
ML Model selection
5G, 6G, visible light communication schemes for ML training and data transfer
Trained human activity recognition model
Store in blockchain for future use
To serve end-user/ system
Federated Learning
On-device training and requires transferring only learning updates
Preserves better privacy
Requires less communication resources due to low size of model weights
Suffers from the challenges of data and system heterogeneity
Convergence is little slow
Centralized learning
Based on moving the training data to a centralized location
Privacy leakage issue
Requires more communication resources for frequently generated data
Fast convergence
Single point of failure issue

FIGURE 1: Overview of the role of machine learning in human activity recognition.

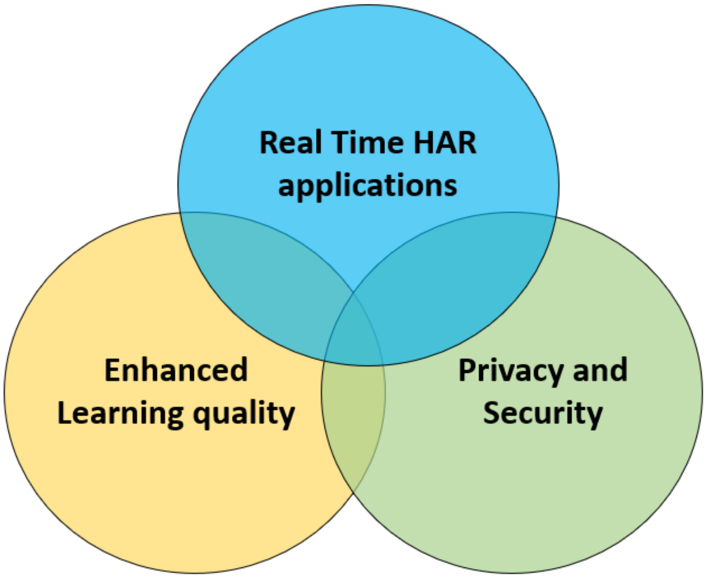

FIGURE 2: The key benefits of FL-based HAR applications.

Finally, we highlight the research challenges and define future directions in FL-HAR. In summary, the main contributions of this paper can be summarized below.

- **FL for HAR, Key Principles and Categories:** We provide readers with essential insights into the key principles and categories of FL as applied HAR.
- **Motivations and Requirements of using FL for HAR:** We begin by identifying the key motivations and highlighting the fundamental requirements that make FL an alternative approach for HAR.
- **Design Aspects, Architecture, and FL Frameworks:** We delve into the design aspects, architectural considerations, and the various FL frameworks that are relevant to HAR applications.
- **Survey of Emerging Applications:** We provide an up-to-date survey of emerging applications of FL within the field of HAR, shedding light on how FL is shaping the landscape of activity recognition.
- **Research Challenges and Future Directions:** We conclude by highlighting the existing research challenges and charting potential future directions in the dynamic field of FL-HAR.

TABLE 1: Summary on FL-related topics and our new contributions

References	Contributions	Limitations
[8]	A survey on the components within FL systems, including aspects such as data distribution, privacy mechanisms, and communication architecture.	The applications of FL for HAR have not been presented.
[9]	A survey on the concepts of FL, including a basic introduction to its architectures and applications.	The applications of FL for HAR have not been presented.
[10]	A survey on the FL hardware, software, platforms, and protocols with a brief introduction to FL-based healthcare use case.	The discussion on FL for HAR is limited
[11]	This survey investigates FL from a software engineering perspective, covering software architecture, development methodologies, and tools.	The applications of FL for HAR have not been presented.
[12]	A survey on the functions and challenges of implementing FL within edge networks, with emphasis on low-latency communication and resource efficiency.	The applications of FL for HAR have not been presented.
[7] [13]	A survey on the FL concept, architecture, and FL-IoT applications, mentioning HAR as a potential application.	The HAR application was partially covered.
[14]	An overview of challenges and prerequisites for implementing FL in the realm of digital health.	The application of FL with HAR has not been explored and discussed.
[15]	A survey on the fundamental architecture of FL models, accompanied by a concise overview of FL's application within the domain of health informatics.	The paper discusses the roles of FL in health informatics in general.
[1] [16]	A survey for FL enabled IoMT	The application within HAR is partially covered.
[17]	A systematic review on FL in the context of EHR data for healthcare applications.	The applications of FL for HAR have not been presented.
[18]	A survey on transfer learning methods in the application domains of HAR.	The FL-based solutions on HAR are partially covered.
This paper	An extensive review of integrating FL within the HAR realm. In particular, first, we highlight the principal motivations and requirements for employing FL in HAR. Subsequently, we present the design aspects, architecture, and FL Frameworks. We also provide an up-to-date survey highlighting emerging applications of FL in HAR, followed by a synthesis of the lessons learned from this survey to offer readers deeper insights into the application of FL in HAR. Lastly, we describe the research challenges and potential future directions in FL-HAR.	-

B. Structure of the Survey

Figure 3 illustrates the organizational structure of this work, where the remainder of this paper is organized as follows. In Section II, we establish the foundational principles and categories of FL as they pertain to HAR, providing the reader with a solid understanding of the core concepts. Following this, in Section III we dive into the motivations behind adopting FL in HAR and outline the essential benefits and requirements that make FL an attractive solution for HAR. Section IV takes a deeper dive into the technical aspects, including architectural considerations and the FL frameworks applicable to HAR. Section V provides an up-to-date survey of FL applications within HAR that includes *health-related activity* and *daily activity*. In Section VI, we discuss the current research challenges and potential future directions for the evolving field of FL-HAR. Finally, Section VII concludes our paper.

II. FL for HAR: Key Principles and Categories

In this section, we start by presenting the basic principles of FL, as well as the different FL types that finds application in HAR, and then we describe several FL aggregation algorithms.

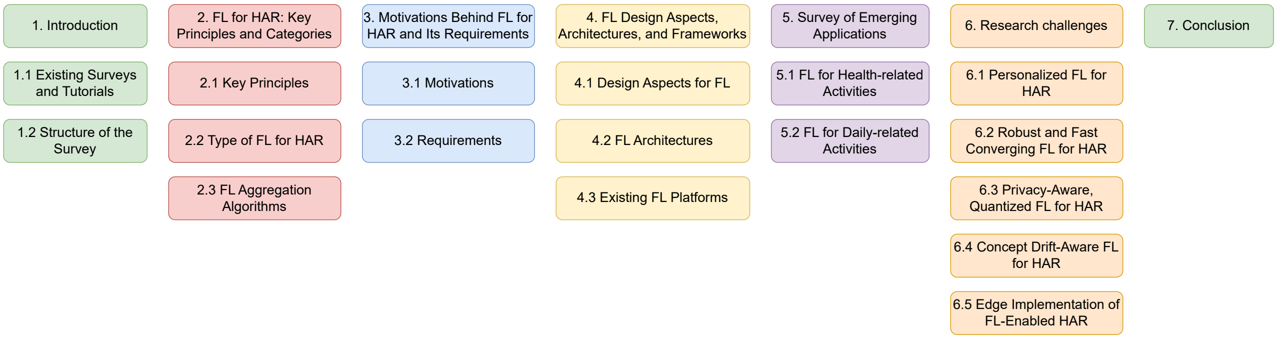
1. Introduction
1.1 Existing Surveys and Tutorials
1.2 Structure of the Survey
2. FL for HAR: Key Principles and Categories
2.1 Key Principles
2.2 Type of FL for HAR
2.3 FL Aggregation Algorithms
3. Motivations Behind FL for HAR and Its Requirements
3.1 Motivations
3.2 Requirements
4. FL Design Aspects, Architectures, and Frameworks
4.1 Design Aspects for FL
4.2 FL Architectures
4.3 Existing FL Platforms
5. Survey of Emerging Applications
5.1 FL for Health-related Activities
5.2 FL for Daily-related Activities
6. Research challenges
6.1 Personalized FL for HAR
6.2 Robust and Fast Converging FL for HAR
6.3 Privacy-Aware, Quantized FL for HAR
6.4 Concept Drift-Aware FL for HAR
6.5 Edge Implementation of FL-Enabled HAR
7. Conclusion

FIGURE 3: The structure of the paper.

A. Key Principles

FL represents a distinctive learning paradigm that enables devices to learn in a collaborative way while preserving data privacy by avoiding data sharing with a central server. It facilitates the training of ML or DL models across multiple devices and servers, thus addressing concerns such as privacy and cost reduction inherent in centralized ML methods [19]. In particular, the FL process consists of two primary steps: local learning and model transmission [20]. Initially, the FL server randomly selects some clients for participation, sending the global model to them. Each client performs local training using its data and then transmits its updated model back to the FL server for global aggregation. This iterative process continues until the performance of the model meets the predefined criteria [7]. This demonstrates how devices may use other devices' data to their advantage through FL without having to send their own private information.

In fact, several aggregation methods have been devised for FL, each with its strengths (Subsection C). Among these, Federated Averaging (FedAvg) stands out due to its simplicity, effectiveness, and robustness [5]. The FedAvg algorithm averages the weights of models trained on local datasets to create a global model. There are also other algorithms and approaches as well such as *FedProx* [21], and *Qffedavg* [22]. The choice of which one to use may depend on the specific requirements of the FL scenario, such as the need for more advanced privacy measures, dealing with unbalanced or non-independent and identically distributed (non-IID) data, or constraints on communication efficiency.

B. Type of FL for HAR

There are three types of FL: Federated Transfer Learning (TFL), Vertical Federated Learning (VFL), and Horizontal Federated Learning (HFL). Each of these FL types has its own set of advantages and is suitable for different scenarios depending on the distribution and privacy requirements of the data.

- *Horizontal Federated Learning (HFL):* In HFL, several clients hold different samples of data, but the feature space of the data is the same. In order to ensure that no raw data needs to leave the individual clients, the clients work together to train a global model. For example, several hospitals may have patient records with the same feature space (attributes), but the specific patients (samples) differ from hospital to hospital [16].
- *Vertical Federated Learning (VFL):* Unlike HFL, VFL is known as feature-based FL, clients may have different sets of features for the same or overlapping data samples. For example, one hospital might have patient demographics and another has their lab results, but both sets pertain to the same group of patients. VFL allows for model training using all available features without sharing the raw data between clients.
- *Federated Transfer Learning (TFL):* FTL extends FL's capabilities to scenarios where data between clients may vary both in feature space and sample space. The concept involves transferring knowledge from one domain to another, allowing for learning across diverse datasets. Even when the characteristics or samples are different, customers with less data can still benefit from learning from those with more data, which can be advantageous in certain instances.

According to Google [23] research, FL can be divided into two categories: cross-silo FL and cross-device FL, depending on the number of client nodes and data availability.

- *Cross-silo FL:* Cross-silo FL usually involves a limited number of clients, 2-100 devices, which are generally easily identifiable and are readily available for training. These silos may represent various entities or departments within a singular organization, each holding extensive datasets. Training data in this context can be categorized into horizontal or vertical learning. However, in this context, computation and communication problems frequently occur. To protect the confidentiality of each client's data, encryption techniques are frequently used, as seen in vertical and transfer learning implementations. An example of this is the work in [24], which utilizes the FATE framework to exhibit cross-silo FL with homomorphic encryption, proposing a batch encryption algorithm based on gradient quantization [25] based batch encryption algorithm to minimize computation and communication overhead in the FL environment.

- *Cross-device FL:* Cross-device FL refers to the FL approach that involves a large pool of clients in the same domain who share a common interest in the global model. Clients, typically individual users and their personal devices, are usually connected via unstable networks, and their participation in training rounds is generally random. Compared to the cross-silo FL, cross-device FL involves more frequent communication rounds but is more lightweight, and participants, less trusted, demand more robust privacy-preserving techniques. Resource allocation strategies, such as client selection/importance [26] and device scheduling [27] are used to choose updates from more beneficial clients, much like data partitioning in HFL. To encourage consumers to participate in FL, incentive systems such as game theory [28] are developed. Cross-device FL is especially well suited for use cases with a large number of clients, such as mobile apps or the cloud Edge IoT continuum (CEI) [29].

C. FL Aggregation Algorithms

Aggregation algorithms are crucial in FL because they determine how the model updates from the local models on the client devices. Depending on the specific objectives, which may include safeguarding user privacy, enhancing convergence speed, and mitigating the impact of fraudulent participants, a diverse array of aggregation algorithms is employed. Each of these strategies comes with its own set of advantages and drawbacks, making some more suitable for particular contexts within the FL realm. In the following, we start describing the most well-known aggregation approaches; then we overview the proposed FL solutions and how they use/combine them.

1) Different Approaches for FL Aggregation

- *Average Aggregation*: This is the original and widely used approach. In this approach, the server computes the average value of the updates received to handle the incoming messages (model updates, parameters, or gradients). Let the number of participating clients be N, and their individual updates at time t be $w_{i,t}$, the aggregate update w_{t+1} is determined as follows [5]:

$$w_{t+1} = \frac{1}{N}\sum_{i=1}^{N} w_{t,i}. \tag{1}$$

- *Stochastic Gradient Aggregation*: Similar to the Average Aggregation, but it takes an average of the gradients instead of model updates [30]. Clients compute the gradients based on their local data and send them to the server, which updates the global model using a learning rate, as in:

$$w_{t+1} = w_t - \frac{\eta}{N}\sum_{i=1}^{N} \nabla f_i(w_t), \tag{2}$$

where η is the learning rate and $\nabla f_i(w_t)$ is the gradient computed by client i.

- *Clipped Average Aggregation*: Like in the previous technique, the messages are averaged, but before calculating the average, the model changes are restricted to a predefined range. This method helps mitigate the influence of outliers and potentially malicious clients that could transmit substantial and malevolent updates [31]. The parameter update evolves as follows:

$$w_{t+1} = \frac{1}{N} \times \sum_{i=0}^{N} clip(w_{i,t}, c), \tag{3}$$

where $clip(x, c)$ is a function that clips the values of x in a range of $[-c, c]$, and c is the designed clipping threshold.

- *Secure Aggregation*: Techniques such as secure enclaves, secure multiparty computation (SMPC), and homomorphic encryption can be used to increase the security of FL. These methods serve to maintain client data confidentiality throughout the aggregation process, a crucial consideration in environments where data privacy is highly prioritized [32]. The secure aggregation approach consists of the integration of an aggregation algorithm with security techniques, where the server can only calculate the aggregate update and cannot access the individual model updates given by the devices. Among these secure aggregation algorithms, one of the most prominent is the Differential Privacy (DP) aggregation algorithm, which introduces a distinctive approach to the integration of client results and is introduced hereafter.
- *Differential Privacy Average Aggregation*: To protect the privacy of client data, an additional layer of DP is added during the aggregation step. Specifically, before transmitting its model update to the server, each client adds random noise to it. The server then combines these updates with random noise to generate the final model [33]. The level of noise incorporated into each update is adjusted to strike a balance between preserving privacy and ensuring model accuracy. If we designate n_i to represent a random noise vector drawn from a Laplace distribution with a scaling parameter b, and this parameter b corresponds to the privacy budget, the aggregated update w_{t+1} results as follows:

$$w_{t+1} = \frac{1}{N} \times \sum_{i=1}^{N} (w_{i,t} + b \cdot n_i), \tag{4}$$

- *Weighted Aggregation*: The server evaluates each client's input in the global model update, taking into account factors like client performance, device type, network connectivity quality, and data similarity to the global distribution. This approach aims to assign greater importance to clients who demonstrate greater reliability or representativeness, thus enhancing the overall accuracy of the model [34]. The aggregate update is computed as:

$$w_{t+1} = \frac{\sum_{i=1}^{N} a_i \times w_{i,t}}{\sum_{i=0}^{N} a_i}, \tag{5}$$

where a_i is the corresponding weight of the client i and w_i its individual updates.

- *Momentum Aggregation*: This approach addresses the issue of slow convergence in FL. Each client maintains a "momentum" term that characterizes the historical trajectory of the model adjustments. Prior to transmitting a fresh update to the server, this momentum term is incorporated into the update. The server accumulates these augmented updates, complete with the momentum term, to construct the ultimate model. This procedure accelerates the convergence of the model [35].
- *Bayesian Aggregation*: The server combines model updates from multiple clients using Bayesian inference, a well-known method that accommodates uncertainty in model parameters. This approach aids in diminishing overfitting and enhancing the model's capacity to generalize to a broader range of data.
- *Adversarial Aggregation*: The server uses various methods to identify and counteract the influence of clients who submit fraudulent changes to the model. These methods encompass techniques such as outlier rejection, model-based anomaly detection, and the use of secure enclaves [36].
- *Quantization Aggregation*: Before transmission to the server for aggregation, the model updates are quantized into a lower-bit format. This process aims to reduce the volume of data to be transferred and improve communication efficiency [37].
- *Hierarchical Aggregation*: The aggregation process is executed across multiple levels of a hierarchical structure. This strategy minimizes communication overhead by performing localized aggregations at lower hierarchy levels before forwarding the results to higher levels [38].
- *Personalized Aggregation*: Throughout the aggregation process, this method takes into account the distinct attributes of each client's data. This approach ensures that the global model is updated in a manner that is best suited to each client's data while protecting data privacy [39].
- *Ensemble-Based Aggregation*: The model is trained on various groups of clients known as ensembles, and the resultant models are harmonized to generate the final model. Each ensemble might involve different subsets of the data, and these ensemble models are subsequently combined to form the final model. This approach can effectively reduce the influence of non-IID data and improve model accuracy.

These algorithms address various challenges in federated learning, such as data heterogeneity, communication efficiency, privacy preservation, and robustness to system variability. The aggregation algorithm choice depends on the federated learning scenario's specific requirements and constraints. We summarize the main pros and cons of these approaches in Table 2, highlighting the use cases recommended for each one. As detailed in the following subsection, many solutions have been proposed on top of these algorithms.

2) Solutions and Implementations for FL Aggregation

As numerous implementations of FL aggregation algorithms are available in the literature, we list some of the most well-known solutions in FL in Table 3.

Federated Averaging (FedAvg) is among the aggregation methods in FL that are used most frequently [5, 12]. First proposed by Google, this approach involves training local models on client devices using their respective data. Subsequently, the model updates (gradients) from each client are sent to a central server, which aggregates these updates by averaging them to update the global model. From 2020, we can see a boost in FL solutions, for example, *Federated Proximal Gradient Descent (FedProx)* is an extension of FedAvg that includes a regularization term to encourage similarity between the global and local models by introducing a proximal term that penalizes divergence of these terms [44]. This solution can handle heterogeneous data and system environments and helps mitigate issues like model poisoning and non-IID data. Similarly, the authors of [45] created the FedMA algorithm, which matches and averages hidden components with comparable feature extraction signatures to build the global model layer by layer. FedMA can surpass classical FL algorithms in handling real-world datasets and simultaneously reduces the total communication overhead. Furthermore, in [42], the authors introduced various FL models that incorporate different adaptive optimization techniques, such as YOGI, ADAGRAD, and ADAM. They conducted an analysis of the convergence of these models under the influence of heterogeneous data within general non-convex scenarios. The results of their research confirmed the viability of these models in accelerating convergence in the context of FL.

In the same context, [46] explored the Analog Gradient Aggregation (AGA) method as a solution to address the limitations of communication resources in FL applications. The solution introduces novel communication and learning strategies that aim to improve the quality of gradient aggregation and accelerate the convergence rate. Furthermore, in [47], the authors presented a low complexity method designed to protect user privacy while demanding considerably fewer computational and communication resources.

TABLE 2: Comparative analysis of different FL aggregation algorithms.

FL Algorithm	Advantages	Disadvantages	HAR Use Case
Average Aggregation	Simple to implement and widely used; Efficient communication and effective for similar data distributions	Sensitive to outliers; Performance degrades with highly heterogeneous data	General-purpose; Works well when HAR data distributions are relatively similar
Stochastic Gradient Aggregation	Precise gradient updates; Efficient for small learning rates	Slower convergence for large models	When more granular control over the learning process is required; More suited for asynchronous FL
Clipped Average Aggregation	Mitigates influence of outliers; Enhances robustness to malicious updates	May require careful tuning of clipping range; Potential loss of information	Scenarios with potential outliers, malicious clients, or HAR data distributions are highly heterogeneous; Enhances security and robustness
Secure Aggregation	Ensures privacy of individual updates; Protects data confidentiality	Computational overhead due to encryption/decryption; Complex implementation	Privacy-preserving scenarios; Sensitive data environments
Differential Privacy Average Aggregation	Enhances privacy by adding noise; Protects client data	Potential trade-off between privacy and model accuracy; Added noise may reduce accuracy	Scenarios requiring strong privacy guarantees; Very sensitive data environments
Weighted Aggregation	Handles data and system heterogeneity; Mitigates the impact of stragglers	Additional complexity due to proximal term; Requires careful tuning of proximal term	Scenarios with heterogeneous data and client computational capabilities
Momentum Aggregation	Accelerates model convergence; Maintains historical trajectory of updates	Added complexity in maintaining momentum terms; Potential for overshooting	Scenarios requiring faster convergence; Large-scale federated learning with slow convergence
Bayesian Aggregation	Reduces overfitting; Handles uncertainty in model parameters	Computationally intensive; Complex implementation	Scenarios needing better generalization; Environments with high uncertainty in data
Adversarial Aggregation	Detects and mitigates influence of malicious clients; Enhances security	Requires robust detection methods; Potential false positives/ negatives	Security-sensitive scenarios; Environments prone to adversarial attacks
Quantization Aggregation	Reduces data transfer volume; improves communication efficiency	Potential loss of information due to quantization; may affect model accuracy	Scenarios with limited communication bandwidth; large-scale federated learning
Hierarchical Aggregation	Minimizes communication overhead; Efficient aggregation at multiple levels	Added complexity in hierarchical structure; Potential latency in multi-level aggregation	Large-scale federated learning; Scenarios with hierarchical client organization; Challenged or high-latency networks
Personalized Aggregation	Tailors global model to client-specific data; Enhances individual client performance	Complex to implement; Requires handling of diverse data characteristics	Scenarios with highly diverse data across clients; Personalized model performance
Ensemble-Based Aggregation	Reduces influence of non-IID data; Improves model accuracy	Computationally intensive; Requires managing multiple ensembles	Scenarios with non-IID data; Improves accuracy and robustness of final model

TABLE 3: State of the art of FL algorithms and their used aggregation approach.

Reference	Year	Solution Name (If present)	Aggregation Approach
[5]	2017	FedAvg	Averaging Aggregation
[40]	2017	–	Secure Aggregation
[41]	2020	SCAFFOLD	Secure Aggregation
[42]	2020	FedOPT	Weighted Aggregation
[42]	2020	FedADAGRAD	Differential Privacy Average Aggregation
[42]	2020	FedYOGI	Personalized Aggregation
[43]	2020	FedBoost	Ensemble-Based Aggregation
[44]	2020	FedProx	Weighted Aggregation
[45]	2020	FedMA	Personalized Aggregation
[46]	2020	–	Stochastic Gradient & Personalized Aggregation
[47]	2020	–	Secure Aggregation
[48]	2020	–	Personalized Aggregation
[49]	2020	LAQ	Quantization Aggregation
[50]	2020	SAFA	Secure Aggregation
[51]	2021	FedDist	Weighted Aggregation
[37]	2021	FedHQ	Quantization Aggregation
[52]	2021	FAIR	Personalized Aggregation
[53]	2021	FedPSO	Ensemble-Based Aggregation
[54]	2021	MHAT	Personalized Aggregation
[55]	2021	–	Secure Aggregation
[56]	2021	–	Weighted Aggregation
[57]	2021	SEAR	Secure Aggregation
[58]	2021	Turbo-Aggregate	Secure & Personalized Aggregation
[59]	2022	EPPDA	Secure Aggregation
[60]	2022	FedBuff	Ensemble-Based Aggregation
[61]	2022	HeteroSAg	Secure & Quantized Aggregation
[62]	2022	RFA	Averaging Aggregation (geometric median)
[63]	2022	LightSecAgg	Secure Aggregation
[38]	2023	P4FL	Hierarchical & Quantized Aggregation
[64]	2023	–	Weighted Aggregation
[65]	2023	FairFed	Weighted Aggregation

The work in [49] devised adaptive communication of quantized gradients, where clients quantize gradients and select transmission of more informative quantized gradients while reusing previous gradient information. This approach results in a "lazy" worker-server communication for the Lazily Aggregated Quantized (LAQ) gradient approach. This model exhibits a substantial reduction in communication overhead. In [50], the authors introduced a semi-synchronous FL protocol named SAFA, with the goal of enhancing the convergence rate in heterogeneous FL scenarios. The distribution of models, client selection, and global aggregation have been designed to mitigate the adverse impacts of stragglers, crashes, and outdated model versions. SAFA effectively shortens the duration of interconnection rounds, minimizes the wastage of local resources, and enhances the accuracy of the global model while maintaining communication costs acceptable.

FedDist is a new FL aggregation technique that detects client dissimilarities to alter its architecture [51]. By using this approach, the model's generalization skills are preserved, but its specificity and personalization are improved. As an alternative, FedHQ [37] speeds up convergence by dynamically determining the right weight for continuing aggregation by computing and adding the quantization error during the local model update. Likewise, FAIR [52] comprises three key components: learning quality estimation, which leverages historical data to estimate the quality of user learning; quality-aware incentive mechanism, which restructures the auction problem to incentivize user engagement with high learning quality; model aggregation, in which only the best models are incorporated to enhance the global model. Federated Particle Swarm Optimization (FedPSO) [53] exhibits increased robustness under unstable network conditions by altering the data clients transmit to servers. Instead of sending extensive weights of local models, FedPSO transmits score values. This can reduce network overhead and traffic volume.

To withstand attacks in FL, the authors of [57] propose the Secure and Efficient Aggregation Framework (SEAR), a Byzantine-robust model. To defend locally learned client models from Byzantine attacks, this approach makes use of Intel Software Guard Extensions (SGX). The authors suggest using two data storage modes to effectively execute the aggregation methods given the memory limitations inside the concurrent trustworthy Intel SGX memory.

Efficient privacy-preserving data aggregation (EPPDA) [59] relies on secret sharing and incorporates an effective fault tolerance strategy to handle user disconnections. The authors conducted tests on their model to demonstrate its resilience against both reverse attacks and disruptions in user connections. In [60], the authors introduced a new FL model named Federated Buffered Asynchronous Aggregation (FedBuff). FedBuff operates independently of the optimizer choice and combines the benefits of synchronous and asynchronous FL, being more efficient than synchronous FL and more efficient than asynchronous FL. In fact, clients engage in asynchronous training and communication with the server. However, unlike typical asynchronous approaches, the server aggregates client updates within a secure buffer before executing the server update, utilizing technologies such as Trusted Execution Environments (TEEs). In order to provide safe aggregation with heterogeneous quantization, HeteroSAg [61] splits updates from the client model into segments and groups the network into segments. Instead of aggregating at the local model level, aggregation is applied to these particular segments with coordinated collaboration among users. By adapting to their available communication resources, edge users can achieve a more balanced trade-off between communication time and training accuracy using this strategy. Furthermore, experiments showed that HeteroSAg is resistant to Byzantine attacks. LightSecAgg [63] is a method based on reconstructing the aggregate mask of active users using “mask coding/decoding” instead of random-seed reconstruction of the dropped users. LightSecAgg reduces overhead for resilience against lost users and offers a modular system design with optimized parallelization on devices, leading to a scalable implementation that enhances the speed of concurrent data exchange.

P4FL [38] has recently proposed a hierarchical FL technique that programs P4 switches to calculate intermediate aggregations of client parameters using the network programmability paradigm. This approach can greatly minimize the cost of communication between clients and the server when used in combination with model quantization. The authors in [64] provide an asynchronous FL architecture with periodic aggregation, also to address the straggler effect, with a similar goal of minimising channel impacts. The study examines the importance of reducing the bias and variance of the aggregated model updates in considering the limited wireless communication resources with HAR applications. Then it suggests a scheduling policy that takes into account both channel quality and the user device representation of the training data. Specifically, learning performance in an asynchronous FL environment can be greatly improved by the suggested “age-aware” aggregate weighting. FairFed [65] attempts to address the fairness problem in FL in different demographic groups, for example, in healthcare and recruitment. FairFed is centralized and agnostic to the applied local debiasing, enabling flexible use of different local debiasing methods across clients. Clients work together with the server to adapt the model aggregation weights. These weights depend on the disparity between the global fairness evaluation (computed over the complete dataset) and the local fairness evaluation at each client. They tend to favor clients whose local measures align with the global measures.

III. Motivations Behind FL for HAR and Its Requirements

This section explains the main motivations and provides detailed requirements for systems that employ FL in HAR.

A. Motivations

We first address the primary drawbacks of the available HAR solutions and then address the benefits that FL may provide for HAR to effectively support its use.

1) Limitations of Current HAR

- *Privacy Concerns:* The use of open data sharing with the cloud or data centers in the deployment of centralized ML-based techniques enables HAR to expose data to privacy threats. In fact, external entities such as cloud service providers could obtain control over data and change patterns without the need for explicit consent from the user [66], or criminal actors could obtain unauthorized access to the central entity to extract data. These inefficiencies may result in serious problems with data leaks that compromise user confidentiality. Although cloud servers have strong computational capabilities that enable effective data training and analysis, there are significant privacy hazards associated with such a centralized ML-based solution for HAR [67].
- *Data availability:* The main challenge to successful implementation of HAR systems is the scarcity of extensive and reliable data sets. Large volumes of diverse, high-quality data are needed to train models that can accurately recognize and categorize a wide range of human activities [1]. The process of data collection can be both time-consuming and expensive, as it often involves monitoring and recording individuals as they perform a variety of activities [68]. These datasets also need to account for the considerable variation in how several people do the same task. It can also be challenging to get information about unusual or rare activities. Finally, privacy concerns can further complicate the process of

collecting HAR data sets [69]. The quantity and kind of data that can be gathered for HAR purposes may be severely limited due to concerns about how these data might be used, which leads to significant limitations on the amount and type of data that can be collected for HAR purposes.

- *Limited HAR performance:* Large-scale, diversified dataset availability is critical to the functioning of HAR systems. The predictive performance of these systems frequently suffers from the lack of large datasets that cover a wide range of human activities [16]. HAR systems that have been trained on sparse or homogeneous datasets, in particular, may not be able to generalize to a variety of situations in life and may even have their robustness compromised. This could lead to reduced accuracy when encountering unrepresented or underrepresented activities in the collected data. Predictions may become skewed if certain demographic groups are not well represented in the dataset. For instance, if the majority of the data in the dataset comes from young adults, the HAR system might not be able to distinguish between tasks carried out by people who are older or younger, or who have different physical capacities. Moreover, a HAR system can overfit the training data [70] [71] if it is trained on a small dataset. Because the system has effectively memorized the training data rather than learning to generalize from it, it will perform well on the training data but badly on fresh, unknown data.
- *High cost of data storage and training:* For centralized ML, data must be processed and kept in one single location, which is typically a cloud-based system or a high-capacity server. Numerous expenses and difficulties are associated with this centralized strategy, including training and storage costs [72]. In particular, HAR systems often make use of substantial amounts of data gathered from several sensors. Large amounts of storage space are required for the central storage of this enormous quantity of data, which can be costly. Furthermore, large computational resources are needed for ML, particularly DL model training for HAR [73]. The effectiveness and speed of the training process are directly impacted by the power of these resources. The requirement for high-performance technology, such as strong CPUs or GPUs, can significantly raise expenses in a centralized paradigm.
- *Communication Cost:* For HAR systems, a centralized training method can result in significant communication costs due to data transfer, bandwidth needs, and energy use [74]. Specifically, for centralized models, all gathered data must be moved to a single place to be used for training. Significant data transfer volumes, increased network utilization, and related expenses can result from this, depending on the volume and complexity of the data, as well as the number of devices [75]. Additionally, a large amount of bandwidth is needed to transfer training data to the central server regularly and receive updates and results in return. This may lead to more network congestion and increased communication expenses, particularly in places where bandwidth is scarce or the cost per data unit is high. Last but not least, transmitting data over a network also consumes energy [20]. This includes not only the energy used by devices to send and receive data, but also the energy used by the network infrastructure.

2) Potential Impact of FL on Future HAR Applications

By leveraging the presented concepts, FL presents several advantages that can significantly enhance HAR systems, as detailed below.

- *Data privacy improvement:* FL improves the protection of user data by keeping the original data on the local device and only sharing changes to the model parameters. This is crucial for HAR applications because user activity data may contain sensitive and private information. Because FL protects user privacy, more users can decide to share their data, which will increase the variety and general quality of the data used for learning.
- *Latency:* Since the data must be sent to a central server for processing, there is generally a delay (latency) associated with centralized learning [76]. Time-sensitive applications, such as medical emergency detection in HAR systems, may have problems due to this latency. FL drastically lowers latency by processing data locally on each device. Decisions may be made more quickly and effectively, improving the HAR system's real-time responsiveness, as each device can train the model and make predictions based on its data without waiting for the server model.
- *Scalability:* Because of the volume of data that must be sent and processed, a centralized ML model may find it difficult to scale efficiently as the number of devices increases. On the other hand, FL makes scaling to a large number of devices much easier by distributing the learning process among the devices themselves. Due to this, FL is a desirable choice for extensive HAR applications, such as those seen in large healthcare systems or smart cities.
- *Less Dependence on Centralized Infrastructure:* By performing learning on the devices themselves or the edge servers, FL reduces the reliance on powerful centralized servers for computation. This can be a significant benefit for HAR systems, particularly in situations where there may be irregular or restricted access to such central resources. This benefit contributes to enhancing the performance and adaptability of HAR applications in diverse operational scenarios.

B. Requirements

To fully leverage the capabilities of FL for HAR within smart healthcare environments, certain key considerations must be addressed, as outlined below.

- *Data Representation:* Interpreting data from different sensors, such as gyroscopes, accelerometers, and even vision-based sensors, is part of HAR. To make the data from these sensors usable for model training, they must be represented and preprocessed. Data cleaning, normalization, segmentation, feature extraction, and data labeling may all be necessary for this. It is critical to extract complete characteristics from the data in the setting of HAR. The research by [66] suggests using a Perceptive Extraction Network (PEN) as a solution to this problem. For each user, the PEN acts as a feature extractor, efficiently processing and analyzing the sensor data to extract relevant information. Nevertheless, an issue presents itself when local device data is frequently unlabeled. Accessing devices to label their data can be a challenging and impractical process due to the nature of FL, where devices are often outside human reach [77]. Solutions to this challenge are being explored in the research domain, with studies like [78] focusing on developing practical methods to handle these kinds of circumstance in FL. Dealing with non-IID data is a considerable problem when preparing FL datasets in HAR. This issue could lead to divergent behavior during FL training. Several strategies are essential to handle non-IID issues and ensure effective training in FL-based smart healthcare. To alleviate the negative effects of non-IID data, one way to provide more representative and balanced data for each client's local model training is to establish extra subsets of datasets that may be distributed evenly among clients. By developing a more comprehensive and resilient global model, this method can improve FL's efficacy in smart healthcare applications. In conclusion, effective data processing and representation, as well as managing unlabeled and non-IID data, are essential components of using FL for HAR. In several real-world applications, more study and development in these fields can greatly enhance the overall performance and reliability of HAR systems.
- *Trusted Server:* In FL processes, a central server is essential because it aggregates the gradients from clients to construct the global model during each communication round. Despite FL's ability to preserve user privacy by allowing them to retain their data locally during training, research has shown that model updates may still contain HAR-related information (e.g., resolution details or particular feature patterns), which could be reconstructed by a curious global server [79]. As a result, confidentiality can be compromised throughout the training process, which may expose FL and discourage health-related organizations from participating in cooperative training. Therefore, an essential need to guarantee reliable FL operations in smart healthcare, especially HAR systems, is to set up a trustworthy server to manage data training and model aggregation. This server must provide computation services that ensure a transparent and reliable model aggregation, aligning with agreements made between the service provider and healthcare organizations, e.g., local hospitals. Trust is especially important in thisspace because, to deliver trustworthy FL-based healthcare services, computations carried out outside of the data sources must be trusted. This is because data about human behavior are extremely sensitive. To bolster trust in the server, recent research efforts have explored new solutions. These include the development of trustworthy and decentralized servers with blockchain technology [80], which can offer an additional degree of security, or the use of safe aggregation techniques. The objective of these efforts is to increase the reliability of FL operations, specifically in the domain of HAR, by guaranteeing the security and dependability of the server that manages confidential information.
- *Local Computational Capabilities in IoMT devices:* One key consideration in the implementation of FL-based solutions, which is based on the participation of mobile medical devices during training, is the computational capacity of these devices. In fact, to maximize federated health care to its fullest, devices need participate in multiple communication rounds to achieve optimal training performance. However, some medical devices, such as small smartwatches, may find it difficult to maintain constant participation in training due to their limited processing power and short battery life [81]. This is a problem since the final FL model is less effective due to the lack of several devices used throughout the training phase. Collective computing power from several devices is a key factor in improving the effectiveness of health data training. Consequently, the need to design specialized computing hardware for health devices arises. Hardware of this type would ideally increase computing speed while consuming less energy, opening the door to an effective and robust FL-driven HAR system. The difficulty becomes considerably greater when HAR is taken into account. For HAR to process sensor input and train models, a significant amount of computer power is needed. Thus, within the FL framework, HAR in smart health devices is made possible by developments in resource allocation methods and energy-efficient models such as spiking neural networks (SNN) [82].

IV. FL Design Aspects, Architectures, and Frameworks

Here, we explore the key design considerations, architectural components, and the different FL frameworks that form the backbone of FL-HAR implementations. Through this exploration, we aim to provide a thorough understanding of the fundamental structural components necessary for FL to be successfully integrated into HAR systems.

A. Design Aspects for FL

This subsection expounds on some recent architectural paradigms for FL-HAR, with the aim of guiding researchers during the design of a new system. Namely, we present advances into privacy-enhanced FL, delineate the contours of secure FL protocols, explore challenges of constrained resources, examine the dimensions of model personalization, and navigate through the complexities of incentive-aware FL.

- *Privacy-enhanced FL:* The model parameters are transmitted to the server for central aggregation after local training. However, one essential component of FL for HAR is secure aggregation, which ensures that model updates are secretly and securely aggregated on the server. *Secure multiparty computation (SMPC)* is one way to provide secure aggregation [83]. SMPC is a subfield of cryptography that facilitates collaborative computation of a function across multiple parties' inputs while maintaining the privacy of such data. SMPC may be used in the FL context to ensure that the server can only calculate the aggregate update and cannot access the individual model updates given by the devices. Protecting the model updates, which may otherwise reveal details about the local data on each device, gives FL an extra degree of privacy. Another method of secure aggregation in FL is *Differential Privacy (DP)* [33]. Differential privacy involves introducing precisely calibrated noise into the data or computation to give a mathematical assurance of privacy. To introduce differential privacy in FL, one approach involves injecting noise into model updates before transmitting them to the server. This guarantees that even while the update serves to enhance the global model, the server cannot deduce particular details about the local data on a device from the model update.
- *Secure FL:* Strong security measures are necessary for effective implementation of FL for HAR to prevent a variety of possible threats, including inference, backdoor attacks, poisoning, malicious servers, and communication bottlenecks [84] [85]. Different sources, such as aggregation methods, data manipulation, and communication protocols, could lead to these attacks [19]. Several security solutions have been developed in recent years for scenarios that involve smart healthcare. Using a reputation-based strategy [86] is one of these strategies to discourage wrong updates from devices that are not trusted. Carefully selecting trusted devices plays a crucial role in reducing security risks. For instance, a malicious device might introduce false information into its local model, compromising the accuracy of the FL process. Ensuring reliability in device selection becomes particularly critical when training local FL models with low-quality or noise-free data. Another viable approach to protecting federated healthcare is decentralized FL, which addresses distrust concerns related to centralized parameter servers [87] [88]. Furthermore, to maintain the reputation of FL users, blockchain technology has been integrated [86]. Specifically, integrating blockchain into FL settings removes the requirement of a single central server in model aggregation [89] by decentralizing the learning process.
- *Resource-aware FL:* Since the devices used in FL for HAR, such as smartphones or wearable sensors, usually have limited computational resources and battery life [16], effective resource management is crucial. Effective resource management can be achieved by using methods such as quantization, model compression, and asynchronous updates. One typical approach to reduce FL's high communication requirements is to employ asynchronous updates. Devices may deliver updates to the central server without relying on other devices due to this technique [90]. In addition to reducing waiting times and processing power required for simultaneous updates, this asynchronous communication takes into account the various computational capacities and network configurations of various devices. Techniques for model compression can further lower FL's transmission and processing expenses. Reducing the size of the model without substantially compromising its accuracy can be achieved by techniques such as quantization, pruning, and knowledge distillation [91]. Pruning removes extra layers or parameters from the model, whereas knowledge distillation trains a smaller model to mimic the behavior of a larger model. One type of model compression that reduces the accuracy of model parameters is called quantization [92]. For instance, a parameter may have been represented in the original model as a 32-bit floating-point number, but it might have been represented as an 8-bit integer in the quantized model. Quantization saves battery life by reducing the model's memory footprint and computational demands. It also minimizes the volume of data that must be sent during FL. In summary, balancing computational efficiency, communication efficiency, and model accuracy is necessary while building FL for HAR to minimize resource utilization. Each of these methods offers a component that completes the picture in order to reach this equilibrium.
- *Model Personalization:* Customizing the global model to the unique needs of each device can greatly enhance the HAR performance. It describes how a globally trained model is modified to better fit the particular data distribution of a single device or user. In particular, aggregated data from several devices are the input data for models in an FL system. However, since multiple devices have inherent data heterogeneity, it could not run at its best on a single device. Diverse sensor

qualities or environmental circumstances might result in different data patterns for every device. Personalization through model fine-tuning is a popular approach. Through transfer learning, each device can further train or ”fine-tune” the model on its local data once the global model has been developed and distributed to devices [93]. As a result, the model's performance and accuracy on that device are enhanced, since it can more effectively adjust to the unique data distribution of the device. Meta-learning, also referred to as “learning-to-learn,” provides an alternative method of personalization [94] [75]. The model in this framework is taught to quickly adapt to novel challenges with little more instruction.

- *Incentive-aware FL:* In traditional FL methods, the device communicates local model updates to an aggregate server. However, this is not always feasible, as IoT devices frequently have restrictions on processing power, bandwidth, privacy difficulties about personal data, and server trust issues. This unwillingness to share models may impede FL's involvement and general effectiveness in HAR systems. To overcome these obstacles and encourage more FL users to participate, incentive mechanisms must be implemented. According to a recent survey [95], these mechanisms may be distinguished according to a number of factors, such as device reputation, contribution to data, and distribution of resources. The quality and quantity of the data are both taken into consideration by the incentive model. The volume of updates and training samples of the model provided by the device is commonly referred to as quantity [96]. On the other hand, metrics like the Shapely value, which measures each member's contribution in a group environment, are used to evaluate quality. However, the reputation of a device has a big impact on how FL incentive algorithms are designed. Reputation usually indicates a device's ability to provide consistent local updates and high-quality data for training models. Additionally, the resource allocation stage of an incentive program is critical since it deals with allocating computing and communication resources among participating FL users in an ideal manner to improve FL's overall performance.

B. FL Architectures

In this subsection, we examine the various FL architectures that serve as the structural blueprint for HAR systems' FL environment, managing the training process among distributed devices and servers.

1) Centralized FL (CFL)

Among the most commonly used FL architectures is CFL. The CFL architecture serves as a robust foundation for HAR, including a central server and a diverse array of clients, many of which are smart devices, such as smartphones or wearables with sensors. During each training cycle, these client devices interpret their localized data, including accelerometer or gyroscope readings, to independently update the shared model. Every client transmits its model parameters to the central server for aggregation when the local training is finished. Usually, the server combines these updates into a single global model using a particular FL aggregation technique (Subsection C). After a few rounds of local training, this global model is sent again to all participating clients, allowing iterative improvement of the global and personalized (local) models. The CFL architecture plays a critical role in coordinating this distributed learning process, ensuring not only efficient training, but also the security and privacy of sensitive user data. For example, in HAR applications where user-specific motion data is highly sensitive, CFL can provide an extra layer of privacy. [6], for example, uses a CFL approach to train deep learning-based activity predictors. This approach, which works well when the data are IID, is also the most common setting in such environments [97, 98].

2) Hierarchical FL

Given that sensitive data are not disclosed, FL theoretically provides some privacy; yet, there are significant drawbacks, such as data that is not distributed identically or independently (non-IID). Specifically, the non-IID data may result in divergence of the final FL model, which means that the performance of FL-based models in the HAR system is not always guaranteed [82]. Furthermore, FL expects that, for model aggregation, the FL server is located in the cloud. There are several difficulties with using the cloud server as an FL server, including communication costs and time delays [82]. For more granular and effective model aggregation, Hierarchical FL for HAR presents a multi-layered architecture including edge and cloud servers [99]. The middle layer's edge servers act as intermediary aggregation points for the local models that have been trained by wearable smartphones and other client devices. 'Sub-global' model aggregations are performed by these edge servers and forwarded to cloud servers at the top tier for global model aggregation. Similarly to CFL, hierarchical FL-HAR allows client devices to participate in the training of a shared global model without requiring the transmission of raw, sensitive data. Large-scale deployment for HAR [100, 101] is especially well suited for hierarchical FL-HAR due to its hierarchical structure, which offers various benefits such as scalability, enhanced data privacy, and optimized network resource utilization. The authors in [102] used HFL in this situation to optimize the heterogeneous electroencephalography (EEG) signals collected from several devices. This paradigm eliminates the problems commonly seen in previous heterogeneous domain adaptation strategies by having each participant have the roles of both a source and a target domain. The results reveal that the proposed approach delivers a significant performance improvement compared to models trained locally without the benefit of hierarchical FL.

3) Decentralized FL (DFL)

Unlike CFL and hierarchical FL, DFL for HAR eliminates the need for a central server to manage the training process. In DFL, client devices, such as smartphones or wearables, are part of a peer-to-peer (P2P) network, where each device trains local models on their human activity data. During each round of communication, clients exchange and aggregate model updates directly with their neighbors in the P2P network. Without the requirement for central orchestration, an agreement on the global model update may be reached using this straightforward, decentralized method. When a centralized server is undesirable or unfeasible, or when a highly scalable network topology is required, DFL is very helpful. It has already shown promise in HAR, such as in [87], which proposes a fully decentralized FL framework by leveraging two state-of-the-art non-convex decentralized optimizations, i.e., decentralized stochastic gradient descent (DSGD) and decentralized stochastic gradient tracking (DSGT). This approach based on DSGT has the advantage of dealing with non-IID datasets.

A safe and open system for model update exchanges may also be established by integrating DFL with decentralized technologies such as blockchain [103]. Blockchain technology can enhance FL's resilience against poisoning attacks to the model. This integration leads to a secure and decentralized process within the IoT framework. Similarly, in [104], distributed agents utilize a combination of blockchain and homomorphic encryption techniques to aggregate data obtained from physical IoT systems before integrating them into the federation model. Clients might communicate with one another via a blockchain ledger in the DFL-HAR environment, providing a reliable platform to safely collect and update models. [105] is an example of an effective approach to tackle the heterogeneity challenge in FL and generate customized high-quality models for each endpoint. Blockchain-based FL enables smarter simulations, reduces latency, and consumes less power while preserving privacy. This solution offers another immediate advantage: in addition to receiving shared model upgrades, the updated model on the phones is automatically utilized, providing personalized insights based on individual phone usage.

C. Existing FL Platforms

In this subsection, we analyze several popular FL platforms, including their design principles, use cases for HAR development, and their limitations and advantages. Table 4 summarizes their description along with their strengths / weaknesses.

- **TensorFlow Federated:** TensorFlow Federated (TFF), an open source and user-friendly framework spearheaded by Google, serves as a specialized platform for both machine learning and decentralized data operations [106]. Given its integrated secure aggregation algorithms and differential privacy safeguards, TFF is well suited for protecting sensitive user information. This is especially crucial in the IoT environment, where a myriad of interconnected devices demand robust security measures.

  Although TFF offers a research-friendly ecosystem for FL scholars to simulate and test HAR algorithms, it is worth noting that the framework currently faces limitations in real-world deployment scenarios. Furthermore, it lacks native support for PyTorch, which restricts its flexibility for HAR researchers who may prefer to use PyTorch-based programs.
- **PySyft:** PySyft, spearheaded by OpenMined, stands as a pioneering framework designed for FL with strong privacy-preserving features, as noted in [107]. In particular, PySyft uses cutting-edge command chains and tensor representations to provide a singular combination of safe data manipulation and data ownership management. The framework provides a dual approach to security within the scope of HAR by smoothly integrating both Multi-Party Computation (MPC) and Differential Privacy (DP) approaches inside the same architectural construct. This makes PySyft an especially appealing option for HAR researchers and developers who need a high level of data privacy and security when conducting FL investigations. The work in [66] used the encryption method provided by the PySyft framework to securely capture sufficient features from HAR data with FL.

  Although PySyft offers a strong framework for secure and privacy-preserving FL, it is not without limitations. For example, implementing PySyft on resource-constrained devices such as IoT sensors, which are commonly used in HAR applications, might be challenging due to its computational requirements.
- **LEAF:** LEAF, led by Carnegie Mellon University (CMU), serves as a specialized benchmarking framework for FL and has made significant contributions to the field [108]. It provides a comprehensive suite of open-source federated datasets, rigorous evaluation mechanisms, and reference implementations to address real-world challenges. Additionally, LEAF offers ready-made implementations of many FL aggregation techniques, including minibatch SGD, FedAvg, Federated SGD, and SGD.

  However, it is worth noting that LEAF has certain limitations, particularly when considered for HAR applications. Currently, LEAF's support is mostly restricted to the FedAvg algorithm, limiting its utility for researchers interested in exploring alternative FL strategies. In addition, it lacks the built-in capabilities for real-world deployment or intricate simulation environments, which are crucial for HAR, a domain requiring real-time analysis and response.
- **PaddleFL:** Paddle FL is an FL framework proposed by Baidu, built on top of its native DL engine, PaddlePaddle [109]. With a focus on transfer and multitask learn-

TABLE 4: A summary of analysis on the applicability of FL platforms for HAR.

Platform	Creator	HAR support	Weaknesses
TFF [106]	Google	- It has a rich set of tutorials and documentation. - It supports basic privacy-related mechanisms for simulation experiments.	- It is not well-compatible for quick deployment on various embedded devices.
Pysyft [107]	OpenMined	- It supports various encryption mechanisms for HAR applications, such as HE and MPC. - It builds on top of popular machine learning frameworks like PyTorch.	- It might not be as compatible with other ML frameworks.
LEAF [108]	Carnegie Mellon University	- It provides a standardized set of benchmarks and datasets. - It offers reference implementations of popular federated algorithms.	- Focuses only on cross-device FL
PaddleFL [109]	Baidu	- It offers FL implementations to support different verticals like IoT and computer vision libraries. - It allows job scheduling and large-scale distributed learning, powered by Kubernetes.	- Its community is not very active.
IBM FL [110]	IBM	- It is tailored for enterprise applications, making it more suitable for large-scale and real-world implementations. - It offers a variety of connection protocols like Flask web framework, gRPC, and WebSockets.	- It does not yet offer advanced security-related algorithms.
FATE [111]	Webank	- It enables cross-silo data applications. - It demonstrates success in deploying smart healthcare applications.	- Its community is not very active.
FedML [112]	FedML	- It can provide support for many network topologies. - Its API is designed to facilitate the development of new FL algorithms	- It lacks good tutorial support.
FedLab [113]	University of Electronic Science and Technology of China	- It improves communication efficiency, which is often a bottleneck in FL systems. - It facilitates both simulations and real-world deployments.	- Its community is not very active.
OpenFed [114]	FederalLab	- It also features a rich library and a flexible topology design. - It supports automatic topology selection, enabling the decomposition of complex FL scenarios into manageable atomic units.	- It lacks robust measures specifically designed to tackle the unique privacy and security issues inherent to HAR applications.
Flamby [115]	Owkin	- It provides seven datasets that cover different tasks in several application domains and with different data modalities and scales. - It helps to compare FL strategies in a fair and reproducible manner.	- Focuses only on cross-silo FL.
Flower [116]	The University of Cambridge	- It supports a large number of devices. - It offers higher-level abstractions and utilities to enable researchers to easily experiment and implement new solutions.	- Does not handle the data distribution or any mechanisms for model encryption.

ing, it provides FL implementations to serve various industries such as computer vision libraries, natural language processing, and the Internet of Things. Paddle enables full-stack development choices, task scheduling, and Kubernetes-driven large-scale distributed learning. When considering its application for HAR, a few challenges emerge. First, the framework's inherent complexity can be a barrier to quick and easy deployment, which is often critical in HAR scenarios that may involve real-time data analysis on mobile or embedded systems. Moreover, the lack of comprehensive documentation and a relatively small developer community further contribute to its steep learning curve. Lastly, Paddle FL has a primarily domestic focus, being most popular among developers in China, which could limit its applicability and support for global HAR projects.

- **IBM FL:** IBM's enterprise grade FL framework provides engineers with a streamlined setup for rapid deployment of federated devices and experimentation [110]. IBM FL provides robust features suitable for distributed machine learning across devices and data centers.
  Although the framework excels at providing rapid deployment capabilities, it currently falls short in specific areas of security and privacy that are critical for HAR applications. In particular, it does not yet offer advanced security features like DP tailored for DL models, which could be a significant limitation for HAR applications requiring stringent privacy and security measures.
- **FATE:** The Federated AI Technology Enabler (FATE) stands as an open-source FL platform, proposed by Webank's AI Department, which emphasizes secure and collaborative machine learning [111]. FATE aims to revolutionize the AI ecosystem by enabling cross-silo data applications that are both distributed and cooperative, while maintaining rigorous compliance and secu-

rity measures. To this end, FATE incorporates advanced secure computation protocols such as homomorphic encryption (HE) and MPC techniques, which can help enhance the privacy of health state data. In particular, a series of independent studies have proposed FATE-compatible deep neural networks [117].
However, when adapting FATE for HAR, there are some inherent challenges. Given that FATE is an industrial grade platform, installing and configuring multiple devices can be a complex task. This might pose issues for HAR applications that often require quick and easy deployment on a variety of mobile and embedded systems. Furthermore, while FATE offers impressive scalability in general terms, it may still have some limitations when applied to real-time, large-scale HAR scenarios, where immediate data analysis and feedback are crucial.

- **FedML:** FedML serves as both a benchmark and a research-oriented FL library, offering an all-encompassing toolkit to develop new FL algorithms as well as to compare existing ones [112]. Customized algorithms in FedML can be easily implemented using the user-oriented programming interface. The primary advantage of FedML is a TopologyManager that can provide support for many network topologies to implement various FL-based solutions.
  Although FedML offers a robust set of features, it is particularly advantageous for HAR applications that require high levels of privacy and security. However, potential users should be aware that FedML's main focus is research, which may limit its out-of-the-box applicability for some commercial HAR applications.
- **FedLab:** FedLab, developed by the University of Electronic Science and Technology of China (UESTC), is a lightweight open-source framework focused on optimizing FL [113]. It aims to improve both the communication efficiency during model training and the performance of standard federated algorithms. The framework features a user-friendly API and a reliable benchmarking tool designed to support simulations and real-world deployments in federated systems with varying computational and communication constraints. This makes this platform a good candidate for HAR applications, where resource limitations are often a concern.
  However, community support might be limited, making the resolution of specific issues more challenging.
- **OpenFed:** OpenFed, a comprehensive and novel FL framework based on PyTorch, serves as an exceptional toolkit. It also features a rich library and a flexible topology design, making it different from other FL frameworks [114]. OpenFed uniquely supports automatic topology selection, enabling the decomposition of complex FL scenarios into manageable atomic units. This is particularly advantageous for HAR, where varying sensor data and user behaviors can introduce complexities. The framework is capable of implementing standard FL algorithms such as SGD and FedAvg. It offers a variety of configuration options that are highly relevant for HAR scenarios. These include partial activation of local client nodes, dataset partitioning, sampling, and the handling of non-IID data distributions. These features facilitate the development of more robust and efficient HAR models.
  Although the framework includes standard FL algorithms, it may lack robust measures specifically designed to address the unique privacy and security concerns in HAR applications.
- **Flamby:** FL AMple Benchmark of Your cross-silo strategies (Flamby) is an open-source FL dataset suite desgined for cross-silo partitions and focused on healthcare. Flamby serves as a bridge between theory and practice of cross-silo FL [115]. It comprises seven healthcare datasets with natural partitions covering multiple tasks, modalities, and data volumes, where each dataset is also accompanied by baseline training code. In addition, it offers standard FL benchmark algorithms for all data sets. Because of the adaptability and modularity of the framework, researchers can simply download datasets, replicate findings, and use various components for their study within HAR.
  However, the Flamby framework focuses on cross-silo FL, which corresponds to the case of a few reliable clients, each holding a medium to large dataset. This may not be representative of other FL scenarios, such as cross-device FL, where there are many unreliable clients, each holding with small datasets. Therefore, the framework may not be suitable for testing FL strategies in different settings.
- **Flower:** Flower is a FL framework designed to facilitate scalable and heterogeneous FL research, offering a distinct advantage in simulating real-world scenarios typical of cloud environments [116]. Based just on a pair of top-tier GPUs, Flower is able to conduct FL experiments with client sizes up to 15 million. Due to this, Flower is an appropriate choice for HAR applications that need realistic, scalable, and secure FL algorithm evaluations. Moreover, it offers higher-level abstractions and utilities to enable researchers and practitioners to experiment with and implement new solutions. For example, the authors in [118] used Flower to predict the length of stay in the hospital.
  However, despite these features, Flower's limitations lie in its lack of a comprehensive ecosystem and restricted support for a broader array of FL algorithms, which could be crucial for specialized HAR applications.

It is crucial to take into account several variables when selecting an FL framework to make sure it fits your unique demands and specifications. The following are some essential considerations:

- *Supported ML Frameworks:* Check if the framework supports the ML libraries and tools that you are familiar with or prefer to use, such as TensorFlow, PyTorch, or scikit-learn.
- *Aggregation Algorithms:* Investigate the aggregation algorithms provided by the framework. While FedAvg is widely used, different frameworks may offer variations or extensions of this algorithm. Understanding the available aggregation methods is crucial to achieving your learning objectives.
- *Privacy Methods and Security:* Assess the framework's support for privacy-enhancing techniques like encryption, differential privacy, and secure multi-party computation (SMPC). Privacy is a critical issue in FL and the ability to implement robust privacy measures is essential.
- *Supported Devices and Operating Systems:* Ensure that the framework is compatible with the devices and operating systems that you intend to use. FL can involve a wide range of devices, from mobile phones to IoT devices, so compatibility is crucial.
- *Scalability:* Evaluate the ease of integrating your ML models or aggregation algorithms into the framework. A flexible and extensible framework allows you to adapt to changing requirements and experiment with novel approaches.

V. Survey of Emerging Applications

In this section, we present a comprehensive overview of recent surveys on FL-HAR. We categorize the current efforts into two main groups: *activities related to health* and *activities related to daily life*.

A. FL for Health-related Activities

In the realm of healthcare, FL for HAR presents a groundbreaking approach to monitoring and analyzing patient physical activities, offering vital insights into their health conditions. As a result, recent solutions have proposed the integration of FL into health-related activities to offer real-time medical services, generating some positive results, summarized in Table 5.

For example, FL for HAR can aid in early detection of conditions such as Parkinson's disease or assessing risk of falls in elderly patients [129]. This application leverages data from various devices, including smartphones and wearable technology, to train ML models while preserving the privacy of individual users. By aggregating diverse and decentralized data, healthcare providers can gain access to more accurate and personalized information on patient health behaviors and patterns. We illustrate in Figure 4 a sample FL architecture for monitoring health-related activities. The data collected by each wearable device do not necessarily have to be sent to a centralized cloud server; instead, they undergo training via FL across multiple edge servers. What sets FL apart is that each device uses its own data to train a local model. Consequently, only the model parameters obtained from the local model are transmitted to the cloud server to update the shared global model.

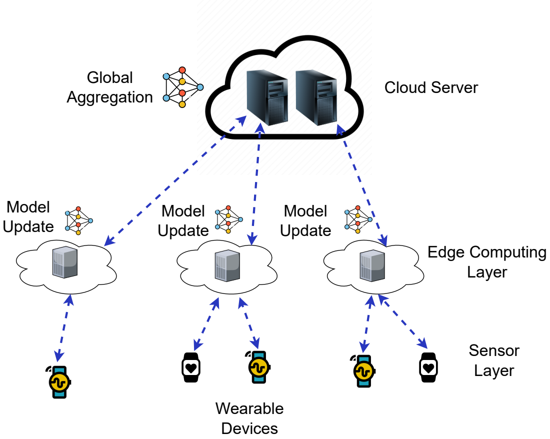

FIGURE 4: A typical architecture for health-related activity predictions using a centralized FL setting.

Furthermore, the concept of combining cloud and edge computing with FL-HAR is stated in [119]. The authors proposed a cloud-edge-based FL framework for home health monitoring, called `FedHome`. It trains individual local models at each home at the network edge, while assigning to the cloud the responsibility of global model aggregation. This training process depends primarily on distributed datasets that can vary from one home to another. `FedHome` uses generative convolutional autoencoders at cloud and edge sites, outperforming several benchmarks in terms of accuracy and communication overhead. In the same direction, a generic FL architecture has been proposed for processing sensor data in HAR by [97]. The proposed solution is based on a federated aggregator trained using private data on edge nodes, demonstrating the versatility and functionality of the FL architecture. In [126], the authors proposed a prototype-guided FL framework, `ProtoHAR`, for handling non-IID data intended for sensor-based HAR. `ProtoHAR` separates the roles of representation and classifiers, corrects local representations using a global activity prototype, and optimizes user-specific classifiers for individualized HAR. This minimizes local model drift and guarantees privacy throughout tailored training. The study demonstrated that `ProtoHAR` outperformed other FL approaches in terms of accuracy and convergence speed. Furthermore, a semi-supervised FL for HAR was explored in [70], focusing on jointly learning deep feature representations of sensor data using autoencoder models. Subsequently, these acquired feature representations are employed for activity recognition within a fully supervised framework using a labeled dataset.

Moreover, another study focused on the utilization of wearable devices to identify and observe patients' activities and movements [121]. FL-based person movement identification, called `FL-PMI` uses DRL to automatically label

TABLE 5: Overview of recent studies on FL for Health-related Activity.

Ref.	FL Type	FL clients	FL aggregator	Datasets	Contribution	Limitations
[119]	Hierarchical FL	In-home sensing devices	Cloud server	MobiAct	A cloud-edge-based FL framework for in-home health monitoring	There is no report about the convergence of the proposed FedHome algorithm.
[97]	CFL	Edge node	FL server	HARUS dataset	A generic FL architecture that processes sensor data for HAR.	The non-IID scenario has not been considered.
[120]	Hierarchical FL	Mobile devices	Cloud server	MHEALTH	A heterogeneous stacked FL architecture to support different architectural local models at patients' devices.	Training latency has not been analyzed.
[121]	Hierarchical FL	Wearable devices	Cloud server	UniMiB SHAR	DRL and Bi-LSTM are used to automatically label the unlabeled data and classify the data, respectively, in a federated way.	Total reliance on the cloud server for global model aggregation.
[122]	CFL	Wearable devices	Cloud server	-	A fog-based IoT platform using FL and blockchain technology to preserve patient data privacy and security within the network.	The computation cost of the proposed platform has not been explored.
[70]	CFL	Edge devices	FL server	OPP, DG, PAMAP2	A collaborative learning of a deep feature representation of sensor data through autoencoder models and these learned feature representation is then used to recognize activities on a labeled dataset in a fully supervised setting.	The model requires labeled data during the training.
[87]	DFL	Hospitals	-	-	Enables communication between nodes and improves communication efficiency for DFL.	A private dataset has been used for performance evaluation.
[123]	CFL	Wearable devices	FL server	MHEALTH	Use GCN-based FL to overcome the issues of privacy preservation and label scarcity in HAR tasks.	The training time has not been discussed.
[124]	CFL	Wearable devices	FL server	UCI, Real-World Dataset	A neuromorphic FL-based model, by integrating the strengths of both LSTM and SNN in a federated setting.	The scalability of the proposed has not been explored.
[125]	CFL	Smart devices	FL server	HARS, HARB	It exploits the advantage of knowledge distillation for distributed training of heterogeneous models.	A public dataset was taken from the training set.
[126]	CFL	Wearable devices	FL server	PAMAP2, USC-HAD, UNIMIB-SHAR, HARBOX	The ProtoHAR separated the roles of representation and classifiers, corrected local representations using a global activity prototype, and optimized user-specific classifiers for individualized HAR.	The proposed model is not able to continuously learn new activity data.
[127]	CFL	Smart devices	FL server	HAR70+, HARTH, MNIST	A novel FUL-based solution for HAR in order to address the challenge of data removal requests under privacy regulations like GDPR.	The scalability of the proposed FUL has not been explored.
[128]	CFL	Smart devices	FL server	UP Fall	A novel solution to address the challenge of data heterogeneity in multimodal fall detection systems by proposing a novel multimodal data fusion method within a FL for HAR.	The complexity of the proposed solution has not been explored.

the unlabeled data and bidirectional long short-term memory (BiLSTM) to extract features and then classify the data. In FL-PMI, the unlabeled data are automatically labeled using the DRL framework. The data were then trained using FL, where the edge server allowed the parameters to be sent separately over the cloud rather than transmitting a substantial quantity of sensor data. Eventually, the data are categorized for different HAR-related procedures via FL-PMI's BiLSTM. The work in [128] addressed the challenge of data heterogeneity in multimodal fall detection systems by proposing a novel multimodal data fusion method within a FL for HAR. Specifically, the method involves the combination of time series data from wearable sensors and visual data from cameras at the input level. The data is transformed into images using the Gramian Angular Field (GAF) method before fusion. In the FL system, each user is treated as a private client, and the fall detection model is trained without sharing user data. On the other hand, to address security issues in wearable IoT devices, [122] proposed a fog-based IoT platform using FL and blockchain technology to preserve patient data privacy and enhance data security within the network.

Since DFL promises to secure FL-HAR and addresses the problem of untrusted parameter servers in CFL, a DFL scheme is proposed in [87], which is a peer-to-peer interaction between health clients made possible by the DFL algorithm. It allows local clients to carry out local updates over multiple iterations. This reduces the time it takes for clients to communicate and exchange parameters, since the models do not need to be sent to a central server that is far away. Federated Graph Neural Networks (GNNs) have also been explored in HAR from sensor measurements. For instance, [123] used a Graph Convolution Network (GCN)-based FL architecture to overcome privacy preservation and label scarcity issues in HAR tasks, building similarity graphs for each user to classify activities in a semi-supervised way.

Although FL offers a degree of privacy for HAR, there are some limitations, particularly when training on low-power and low-computational resource devices such as wearable sensors. The authors in [124] proposed a neuromorphic FL-based model, called S-LSTM, by integrating the strengths of both LSTM and SNN in a federated setting. SNN is an event-driven learning process that significantly reduces energy consumption. The outcomes of the proposed S-LSTM show how much better it can recognize the time patterns within health-related activities while using less energy. Similarly, the authors in [125] explored knowledge distillation for distributed training of heterogeneous models in FL, reducing communication overhead, achieving faster convergence, and lowering the energy cost of FL models. Furthermore, [120] proposed a novel heterogeneous stacked FL architecture supporting heterogeneous architectural client models to overcome the limitation of heterogeneous architectural ensembling in the traditional FL approach.

Despite substantial progress in user privacy protection with FL, challenges persist. GDPR empower users to request data removal. Federated unlearning (FUL) can address this challenge by enabling the selective removal of a client's data from the trained model without retraining from scratch, thus maintaining privacy and efficiency [130]. In this context, the authors in [127] introduced a lightweight FUL method. They used a third-party dataset and Kullback-Leibler divergence (KL divergence) as a loss function to fine-tune the FL model, ensuring the predicted probability distribution on the data to be forgotten aligns with that of the third-party dataset. Additionally, a membership inference evaluation is used to assess the unlearning effectiveness. Experimental results show that this approach achieves unlearning accuracy comparable to traditional retraining methods, with significant computational speedups, thus providing an efficient solution for handling data removal in FL scenarios.

B. FL for Daily-related Activities

In the context of daily life, HAR can be used to understand and optimize individual behaviors and routines. This could include tracking fitness routines, detecting driving habits, or even understanding household activities [129]. By harnessing the power of FL, data from various users can be aggregated to create more robust models without compromising individual privacy [13]. Consequently, recent studies have proposed the integration of FL into daily activities to improve quality of life and have achieved some good results, summarized in Table 6, as follows.

For example, the work in [66] presented an FL system for wearable sensor-based HAR, which is known as HARFLS. With the help of HARFLS, each user can safely and cooperatively complete their activity recognition job. Addressing the challenges posed by non-IID data in FL-based HAR, [67] conducted a comprehensive investigation, shedding light on factors such as diverse subsets of activity and data corruption. To harness valuable features from HAR data while combating statistical heterogeneity, a perceptive extraction network (PEN) was designed, as demonstrated by its superior performance compared to existing methods. Another work focused on federated feature extraction called FedMAR is presented in [132]. It treats the HAR problem associated with each user as a separate learning task. FedMAR framework leverages multimodal wearable data and exhibits rapid adaptability to new individuals. This framework uses an attention module for each client, enabling the learning of both client-specific features and globally correlated features. The work in [141] conducted an evaluation of various FL optimizers, with findings that emphasize the effectiveness of federated averaging for superior global performance. Moreover, the authors in [140] introduced FedCoad, an innovative approach designed to address skewness between different clients in real-world settings. FedCoad utilizes model contrastive learning to align global and local model representations and applies control variates to regularize local model updates. This method aims to build a generalized global model that can be adapted by participating clients without collecting their sensor data. Experimental results show that FedCoad significantly outperforms other methods in skewed dataset settings (non-IID) on benchmark datasets, highlighting its ability to effectively manage data heterogeneity.

Considering the practical aspects and considerations surrounding FL-based HAR, [72] provided a system-level perspective, offering insights into the impact of factors such as sensor location, FL optimizer, and model complexity.

To adapt FL models for heterogeneous devices, [131] dynamically adapt the model layers and model sizes for heterogeneous devices to participate in FL. In particular, the authors proposed FL via Dynamic Layer Sharing, FedDL, a dynamic layer-sharing scheme that learns the similarity among users' model weights to establish the sharing structure and merges models accordingly in a bottom-up layer-wise manner. The objective is to facilitate accurate daily activity recognition by training personalized deep models for users with limited or unbalanced data. The paper also presents a new dataset collected using LiDAR and four real public datasets to evaluate the performance of FedDL. The authors claim that this scheme can improve the accuracy of the FL model, the convergence rate, and the communication overhead of the HAR compared to several state-of-the-art FL-based solutions. In the same direction, the authors in [133] addressed challenges such as concept drift and convergence instability in personalized FL with FedHAR, employing hierarchical attention architecture and unsupervised gradient aggregation. They devised an unsupervised gradient aggregation technique to address challenges related to drift and convergence variability, employing online learning to enhance the process. In particular, FedHAR

TABLE 6: Overview of recent studies on FL for Daily-related Activity.

Ref.	FL Type	FL clients	FL aggregator	Datasets	Contribution	Limitations
[67]	Hierarchical FL	Smart devices	FL server	HHAR	Investigation of the applicability of FL for HAR with non-IID data and the presence of corrupted data.	The scalability of the FL has not been explored.
[66]	Hierarchical FL	Mobile devices	FL server	WISDM, UCI-HAR, Opportunity, PAMAP2	A feature extractor to extract the local features and global relationships from heterogeneous data to address statistical heterogeneity.	The scalability of the proposed solution has not been explored.
[72]	Hierarchical FL	Smart devices	FL server	PAMAP2, JSI-FOS	A study of FL-based HAR under different real-world scenarios, including communication cost/bandwidth efficiency, model complexity, and inaccurate data.	No optimization solution was proposed.
[131]	CFL	Smartphone	Cloud server	UWB, Depth Images, HARBOX-IMU, IMU, LiDAR	A dynamic layer-sharing scheme that learns the similarity among users' model weights to form the sharing structure and merges models accordingly in an iterative, bottom-up layer-wise way.	The same model architecture is used with all devices.
[132]	CFL	Wearable devices	FL server	HHAR, PAMAP2, ExtraSensory, SmartJLU	Using an attention module for each client to learn both client-specific features and globally correlated features while preserving data privacy.	The computation capacity of the devices was not considered during the training process.
[133]	CFL	Wearable devices	FL server	RealWorld, HAR-UCI	A personalized federated HAR framework based on semi-supervised online learning.	The complexity of the model was not discussed.
[134]	Hierarchical FL	Wearable devices	Cloud server	MobiAct, WISDM	With FedCLAR, the local models received by the server are clustered and merged considering the similarity of their weights.	The labeled data needs to be available on each client.
[135]	CFL	Wearable devices	Cloud server	MobiAct, WISDM	A recent hybrid method for HAR combining semi-supervised learning (i.e., active learning) and FL to leverage the strengths of both approaches.	The non-IID data problem has not been considered.
[136]	Hierarchical FL	Wearable devices	Cloud server	MobiAct, WISDM	A combination of FedCLAR and FedAR by proposing Semi-Supervised-FedCLAR Based HAR in order to mitigate both the non-IID and data scarcity problems.	The complexity of the model has not been discussed.
[137]	Hierarchical FL	FL server	Wearable devices	IMU, UWB, FMCW, Depth, WISDM, MobiAct, HARBox	A novel integration of personalized FL with hierarchical clustering.	The latency has not been evaluated.
[138]	CFL	Smartphone	Cloud server	-	Combining VFL+HFL to support heterogeneous data sharing with privacy protection.	The complexity of the model was not analyzed.
[2]	Hierarchical FL	Smart devices	FL server	HHAR, MobiAct, HARBox	A Hybrid-model federated learning mechanism, which allows devices to train model parts suited to their capabilities. It clusters devices based on model similarity to mitigate data heterogeneity impacts and introduces a pairing scheme for effective co-training between high- and low-performance devices.	The complexity of the model has not been discussed.
[139]	CFL	Smart devices	FL server	WISDM	A novel framework that extend the Multi-level FL architecture with three specialized methods tailored to tackle specific heterogeneities: statistical, device, and model.	The proposed framework integrates multiple specialized methods, which require significant computational.
[140]	CFL	Smartphones	FL server	WISDM, Motionsense, HHAR	A federated model contrastive learning to address skewness between different clients in real-world settings.	The unlabeled data have been ignored during the training process.

uses a hierarchical attention architecture to align different level features, employing three main components: a semi-supervised learning loss function to aggregate gradients from all labeled and unlabeled clients; a novel algorithm for computing unsupervised gradients under the consistency training proposition; and an unsupervised gradient aggregation strategy to address the issues of concept drift and convergence instability in online learning. Furthermore, the authors in [2] introduced a hybrid model federated learning mechanism, called Hydra. Hydra employs BranchyNet to create a large-small global hybrid model, enabling devices to train model parts suited to their capabilities. It clusters devices based on model similarity to mitigate data heterogeneity impacts and introduces a pairing scheme for effective co-training between high- and low-performance devices. Additionally, Hydra employs a sample selection approach to enhance co-training efficacy and proposes a Large-to-Small knowledge distillation algorithm to optimize knowledge transfer from large to small models, significantly improving model accuracy. Extensive experiments on three HAR datasets validate Hydra's superior performance compared to state-of-the-art schemes. The work in [139] proposed a personalized Multi-level Federated Learning, PerMl-Fed has been proposed as an innovative framework. This approach extends the Multi-level FL architecture with three specialized methods tailored to tackle specific heterogeneities: statistical, device, and model. The Transfer Multi-level FL model mitigates statistical heterogeneity across multiple FL layers, while the Asynchronous Multi-level FL approach allows asynchronous updates to address device heterogeneity. Additionally, the Deep Mutual Multi-level FL method employs deep mutual learning to overcome model heterogeneity. Evaluations in the WISDM dataset show that PerMl-Fed significantly improves the average precision by 7%, achieving an accuracy range of 84% to 92% in various hierarchical group structures, demonstrating its effectiveness in improving federated learning performance.

Along the statistical (non-IID data) and model heterogeneity, label heterogeneity presents a substantial challenge, especially when each FL device has its own definition of data labels, independently from the definitions in other devices or the central server. This particular type of heterogeneity arises when diverse devices have disparate understandings or classifications of data labels, potentially leading to inconsistent training data across the federated network. Consequently, several works have been proposed to address the problem of heterogeneity in labels between users using model distillation techniques and to demonstrate the validity of the approach with an average increase of 9.153%–11.01% using daily activity recognition datasets [142] and [78]. This underscores the potential for model distillation techniques to not only navigate but also leverage label heterogeneity, providing a robust pathway for improved FL even in the face of varied data definitions across devices.

On the other hand, to build accurate FL-based HAR models, it is essential to find a balance between generalization and personalization. To address this issue, a solution called FedCLAR has been proposed in [134] to generate specialized global models (server-side) for groups of similar users. With FedCLAR, the local models received from the server are clustered taking into account the similarity of their weights. While FedCLAR significantly improves personalization, it relies on the availability of labeled data on each client. However, collecting a large annotated dataset for each client is often impractical due to its time-consuming, costly, and intrusive nature. Furthermore, the work in [135] proposed FedAR, which is a novel hybrid approach for HAR that combines semi-supervised and FL to capitalize on the benefits of both methodologies. Specifically, FedAR integrates active learning and label propagation to semi-automatically annotate the local streams of unlabeled sensor data, while FL is used to build a global activity model in a scalable and privacy-aware manner. The results indicate that the combination of active learning and label propagation yields recognition rates comparable to fully supervised methods. In the same vein, [136] has combined FedCLAR and FedAR to propose Semi-Supervised-FedCLAR Based HAR, called SS-FedCLAR. The objective is to address non-IID and data scarcity problems, and the results show that SS-FedCLAR outperforms FedAR and reaches results close to those of FedCLAR with a limited amount of labeled data.

However, the mentioned studies do not consider possible attacks in the FL setting, which could perturb the training process, e.g., via data poisoning and model poisoning attacks. The work in [137] presents a novel integration of personalized FL with hierarchical clustering, known as FedCHAR. The proposed FedCHAR not only enhances the fairness and accuracy of the model by using similar relationships between users in the benign scenario, but it also improves the robustness of the system by identifying malicious nodes through clustering. The work in [138] proposed a 2D FL framework taking advantage of the VFL and HFL phases to address concerns about unsafe data sharing and inadequate training data in cyberphysical systems. In particular, the solution uses the VFL phase to improve performance by integrating patient features from different devices, and then the server uses the HFL phase to average the global model from different patients.

VI. Research challenges

A. Personalized FL for HAR

How can personalized FL for HAR be enabled that can perform well for various applications? HAR applications generate data that is specific for various applications, groups of users, and geographic areas. Therefore, training a generalized FL model might not work well for HAR. For instance, considering gesture recognition, one can see that ML models trained for a certain geographic location (i.e., supermarket) will not work well for gesture recognition of another geographic location (i.e., hospital or special children's school) because of their different nature. In hospitals, the data will mostly have gestures of sadness. On the other hand, in supermarkets, the gestures of people will have a mostly different nature than sadness. Therefore, there is a need to train FL models for HAR that can perform well in various scenarios. To do so, a personalized FL is needed. One possible way is to use federated meta-learning, which involves sharing a meta-learner instead of a global model [143]. Specifically, the goal of meta-learning is to enable the training of models to learn how to learn. This type of learning will enable faster convergence to the specific HAR scenarios. Another possible solution could be to use the pre-trained models for specific scenarios. Pre-training a model on a large and diverse dataset allows it to capture general patterns of human activity. These pretrained models can then be fine-tuned on specific datasets to adapt to the particularities of various HAR environments [144]. Furthermore, several personalized FL algorithms can be utilized to enhance the performance of HAR models in various applications. These algorithms can be designed to incorporate local adaptations while maintaining the benefits of collaborative learning. For example, MOCHA is an algorithm designed to handle the heterogeneity of data between clients by solving multiple tasks jointly but allowing task-specific adaptations [145]. This approach can be particularly useful for HAR applications where the data of each client could differ significantly from the others.

B. Robust and Fast Converging FL for HAR

How does FL for HAR robust along with fast convergence? FL is based on a single centralized aggregator that will suffer from malfunction if the aggregator stops working. This can happen due to many reasons, such as physical damage or security attacks. To remedy this, one can use the concept of distributed aggregations. However, FL based on distributed aggregation can avoid a single point of failure, but at the cost of high cost in terms of communication resources and implementation. Therefore, a trade-off must be made between robustness and complexity. On the other hand, the convergence of FL is generally slow. Hierarchical aggregations, where multiple levels of aggregators are used, can also speed up convergence by reducing communication latency and distributing the computational load [146]. Additionally, heterogeneity-aware clustering groups clients based on data or computational similarities, optimizing the training process by treating each cluster as a separate federated learning task. This method addresses data heterogeneity and ensures that the model is more robust and better tailored to specific client groups. For example, FedProx [44], a robust FL algorithm that adds a proximal term to local objective functions to handle heterogeneity and improve convergence. Similarly, hierarchical FL frameworks proposed in [147] for edge computing environments reduce communication overhead and accelerate convergence through multilevel aggregations. By employing these advanced techniques and addressing the associated challenges, FL for HAR can achieve robust, fast-converging models that operate efficiently in real-world environments, ensuring effective and adaptive security measures.

C. Privacy-Aware, Quantized FL for HAR

How do we enable FL for privacy-aware HAR along with less communication overhead? As the number of devices is expected to grow exponentially in the foreseeable future, enabling FL for HAR applications will necessitate substantial communication resources. Given that communication resources are limited, it is essential to redesign or improve our systems to accommodate more devices within the FL framework for HAR. One approach is to enhance resource management by optimizing the communication protocols and reducing the size of model updates through quantization schemes. Quantization reduces the communication overhead by compressing the model updates, which significantly lowers the amount of data transmitted during each communication round. However, while FL inherently offers a degree of privacy by keeping data localized on edge devices, it does not fully protect against privacy breaches. Malicious nodes or aggregation servers can still infer sensitive information from model updates shared during the training process. To mitigate this risk, privacy-preserving techniques such as differential privacy and homomorphic encryption can be employed. Differential privacy adds noise to model updates, making it difficult for adversaries to extract meaningful information about individual data points. Homomorphic encryption, on the other hand, allows computations to be performed on encrypted data, ensuring that the raw data remains confidential even during the training process. For example, the work in [148] demonstrated the application of differential privacy in machine learning by incorporating noise into the training process to protect individual data points while maintaining the accuracy of the model. Similarly, an homomorphic encryption, which enables secure computations on encrypted data without revealing the underlying information is proposed in [149]. These techniques can be integrated into FL systems to enhance privacy while minimizing the communication overhead. In addition to these privacy-preserving methods, advanced resource management strategies such as adaptive bandwidth allocation and dynamic compression techniques can further optimize the communication efficiency. By dynamically adjusting the communication parameters based on the network conditions and device capabilities, the FL system can effectively balance the trade-offs between communication overhead and model performance.

D. Concept Drift-Aware FL for HAR

How do we enable FL to update the model in response to concept drift in HAR? Concept drift refers to changes in user behavior (i.e., output). In HAR, the concept drift can be due to many factors, such as changes in lifestyle, seasons, or even cultural changes. Therefore, there is a need to continuously keep the HAR system updated as per concept drift. Since most of the HAR modules will be based on FL in the future, therefore, one must propose concept drift-aware FL algorithms. To address concept drift, FL models must be continuously updated to reflect new data and evolving patterns. This continuous updating can improve the performance of HAR systems, but may also increase communication overhead. Federated Unlearning (FUL) offers a solution to this challenge by balancing performance improvements with communication costs [150]. FUL allows for the selective forgetting of outdated or irrelevant data, ensuring that the model remains relevant without excessive communication overhead. Implementing drift-aware FL concept involves several strategies. An approach is to incorporate adaptive learning rates that adjust based on the detected drift, ensuring that the model quickly adapts to new patterns while minimizing unnecessary updates. Another strategy is to use ensemble methods, where multiple models are trained on different data subsets and combined to provide robust predictions that account for drift. For example, the work in [151] proposed FUL to address the need for model updates in response to concept drift. By selectively removing outdated data and incorporating new information, FUL maintains the relevance of the model while managing communication costs. Furthermore, strategies such as incremental learning [152] can be used to continuously adapt FL models to new data without starting from scratch, thus reducing the communication burden.

E. Edge Implementation of FL-Enabled HAR

How does one efficiently implement FL algorithms for HAR on the network edge? Efficiently implementing FL for HAR on the network edge requires overcoming several challenges related to computational resources and power consumption. The edge, characterized by limited computing and backup power, requires low-complexity schemes to handle complex HAR tasks [153, 154]. Model compression techniques such as quantization, pruning, knowledge distillation, and spiking neural network (SNN) are essential to reduce the complexity and size of FL models, making them suitable for deployment on resource-constrained edge devices. Quantization reduces the precision of the model parameters, which decreases the size of the model and the computational resources required without significantly affecting performance [155]. Pruning involves removing less significant weights from the model, further reducing the computational burden [156]. Knowledge distillation transfers knowledge from a large, complex model (teacher) to a smaller, simpler model (student), thereby retaining performance while reducing the size of the model [157]. SNN is a new generation of neural networks. It is an event-driven learning process and, in turn, significantly reduces energy consumption [158]. These techniques ensure that FL models can be effectively deployed on resource-constrained edge devices, enabling robust and real-time HAR applications.

VII. Conclusion

In this paper, we present the role of FL in enabling privacy-preserving HAR applications. Our findings show that FL not only enhances privacy by keeping data on local devices but also improves the accuracy of the model by leveraging data from diverse sources without sharing raw information. These contributions are crucial to addressing key challenges in HAR, such as data heterogeneity, privacy concerns, and the demand for real-time processing. The practical implications of this work suggest that FL can be a cornerstone in developing scalable, secure, and adaptive HAR systems, especially in environments where data privacy is paramount. Moreover, as FL continues to evolve, it opens up possibilities for deploying HAR applications in healthcare, smart cities, and wearable technologies. However, several challenges remain to be tackled, such as optimizing FL communication protocols, handling non-IID data distributions more effectively, and ensuring model robustness against adversarial attacks. Future research should focus on refining these areas while exploring new strategies, like integrating FL with emerging technologies like edge computing, to create more efficient and reliable HAR systems. These next steps will help solidify FL's role in the future of HAR and other privacy-critical applications.

Acknowledgement

This work was supported by the European Union Horizon-CL4-2021 Research and Innovation Programme under Grant Agreement 101070181 (TALON). The paper reflects the authors' views, and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[2] P. Wang, T. Ouyang, Q. Wu, Q. Huang, J. Gong, and X. Chen, "Hydra: Hybrid-model federated learning for human activity recognition on heterogeneous devices," *Journal of Systems Architecture*, vol. 147, p. 103052, 2024.

[3] S. Ek, F. Portet, P. Lalanda, and G. E. V. Baez, "Evaluating federated learning for human activity recognition," in *Workshop AI for Internet of Things, in conjunction with IJCAI-PRICAI*, 2021.

[4] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, and S. Seshan, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2020, pp. 104–115.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learn-

ing of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[6] S. Zehtabian, S. Khodadadeh, L. Bölöni, and D. Turgut, "Privacy-Preserving Learning of Human Activity Predictors in Smart Environments," in *IEEE INFOCOM-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[7] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[8] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[9] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[10] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[11] 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.

[12] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[13] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[14] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, p. 119, 2020.

[15] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, pp. 1–19, 2021.

[16] O. Aouedi, A. Sacco, K. Piamrat, and G. Marchetto, "Handling Privacy-Sensitive Medical Data With Federated Learning: Challenges and Future Directions," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 790–803, 2022.

[17] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[18] S. G. Dhekane and T. Ploetz, "Transfer learning in human activity recognition: A survey," *arXiv preprint arXiv:2401.10185*, 2024.

[19] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[20] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, 2022.

[21] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[22] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[23] 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.

[24] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning," in *USENIX annual technical conference (ATC '20)*, 2020, pp. 493–506.

[25] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in neural information processing systems*, vol. 30, 2017.

[26] Z. Tao and Q. Li, "eSGD: Communication Efficient Distributed Deep Learning on the Edge," in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[27] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[28] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *IEEE Networking Letters*, vol. 2, no. 1, pp. 23–27, 2019.

[29] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied

federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, 2018.

[30] H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," *Advances in Neural Information Processing Systems (NeurIPS 2020)*, vol. 33, pp. 5332–5344, 2020.

[31] T. Wang, Z. Zheng, and F. Lin, "Federated learning framework based on trimmed mean aggregation rules," *SSRN Electronics Journal*, 2022.

[32] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

[33] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[34] J. Reyes, L. Di Jorio, C. Low-Kam, and M. Kersten-Oertel, "Precision-weighted federated learning," *arXiv preprint arXiv:2107.09627*, 2021.

[35] J. Xu, S. Wang, L. Wang, and A. C.-C. Yao, "FedCM: Federated Learning with Client-level Momentum," *arXiv preprint arXiv:2106.10874*, 2021.

[36] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 634–643.

[37] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6804–6819, 2021.

[38] A. Sacco, A. Angi, G. Marchetto, and F. Esposito, "P4FL: An Architecture for Federating Learning with In-Network Processing," *IEEE Access*, pp. 103 650–103 658, 2023.

[39] X. Ma, J. Zhang, S. Guo, and W. Xu, "Layer-wised model aggregation for personalized federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 092–10 101.

[40] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '17')*. ACM, 2017, pp. 1175–1191.

[41] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 5132–5143.

[42] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[43] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: A Communication-Efficient Algorithm for Federated Learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 3973–3983.

[44] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems 2 (MLSys 2020)*, vol. 2, 2020, pp. 429–450.

[45] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[46] H. Guo, A. Liu, and V. K. Lau, "Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 197–210, 2020.

[47] B. Choi, J.-y. Sohn, D.-J. Han, and J. Moon, "Communication-computation efficient secure aggregation for federated learning," *arXiv preprint arXiv:2012.05433*, 2020.

[48] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23 920–23 935, 2020.

[49] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, "Lazily aggregated quantized gradient innovation for communication-efficient federated learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2031–2044, 2020.

[50] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "SAFA: A Semi-Asynchronous Protocol for Fast Federated Learning With Low Overhead," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 655–668, 2020.

[51] E. Sannara, F. Portet, P. Lalanda, and V. German, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[52] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "FAIR: Quality-Aware Federated Learning with Precise User Incentive and Model Aggregation," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[53] S. Park, Y. Suh, and J. Lee, "FedPSO: Federated learning using particle swarm optimization to reduce communication costs," *Sensors*, vol. 21, no. 2, p. 600,

2021.

[54] L. Hu, H. Yan, L. Li, Z. Pan, X. Liu, and Z. Zhang, "MHAT: An efficient model-heterogenous aggregation training scheme for federated learning," *Information Sciences*, vol. 560, pp. 493–503, 2021.

[55] B. Jeon, S. Ferdous, M. R. Rahman, and A. Walid, "Privacy-preserving decentralized aggregation for federated learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2021, pp. 1–6.

[56] Y. Wang and B. Kantarci, "Reputation-enabled federated learning model aggregation in mobile platforms," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[57] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "SEAR: Secure and Efficient Aggregation for Byzantine-Robust Federated Learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3329–3342, 2021.

[58] J. So, B. Güler, and A. S. Avestimehr, "Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 479–489, 2021.

[59] J. Song, W. Wang, T. R. Gadekallu, J. Cao, and Y. Liu, "EPPDA: An Efficient Privacy-Preserving Data Aggregation Federated Learning Scheme," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 3047–3057, 2022.

[60] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated Learning with Buffered Asynchronous Aggregation," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 3581–3607.

[61] A. R. Elkordy and A. S. Avestimehr, "HeteroSAg: Secure Aggregation With Heterogeneous Quantization in Federated Learning," *IEEE Transactions on Communications*, vol. 70, no. 4, pp. 2372–2386, 2022.

[62] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[63] 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," in *Proceedings of Machine Learning and Systems (MLSys '22)*, vol. 4, 2022, pp. 694–720.

[64] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and Aggregation Design for Asynchronous Federated Learning Over Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 874–886, 2023.

[65] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, and A. S. Avestimehr, "FairFed: Enabling Group Fairness in Federated Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7494–7502.

[66] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, p. 107338, 2021.

[67] K. Sozinov, V. Vlassov, and S. Girdzijauskas, "Human activity recognition using federated learning," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. IEEE, 2018, pp. 1103–1111.

[68] Q.-V. Pham, D. C. Nguyen, T. Huynh-The, W.-J. Hwang, and P. N. Pathirana, "Artificial intelligence (ai) and big data for coronavirus (covid-19) pandemic: a survey on the state-of-the-arts," *IEEE access*, vol. 8, pp. 130 820–130 839, 2020.

[69] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[70] Y. Zhao, H. Liu, H. Li, P. Barnaghi, and H. Haddadi, "Semi-supervised federated learning for activity recognition," *arXiv preprint arXiv:2011.00851*, 2020.

[71] E. Diao, J. Ding, and V. Tarokh, "Semifl: Communication efficient semi-supervised federated learning with unlabeled clients," *arXiv preprint arXiv:2106.01432*, 2021.

[72] S. Kalabakov, B. Jovanovski, D. Denkovski, V. Rakovic, B. Pfitzner, O. Konak, B. Arnrich, and H. Gjoreski, "Federated learning for activity recognition: A system level perspective," *IEEE Access*, 2023.

[73] C. Fang, Y. Guo, N. Wang, and A. Ju, "Highly efficient federated learning with strong privacy preservation in cloud computing," *Computers & Security*, vol. 96, p. 101889, 2020.

[74] M. Uddin, A. Salem, I. Nam, and T. Nadeem, "Wearable sensing framework for human activity monitoring," in *Proceedings of the 2015 workshop on Wearable Systems and Applications*, 2015, pp. 21–26.

[75] O. Aouedi and K. Piamrat, "F-bids: Federated-blending based intrusion detection system," *Pervasive and Mobile Computing*, vol. 89, p. 101750, 2023.

[76] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for uavs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53 841–53 849, 2020.

[77] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Federated semisupervised learning for attack detection in industrial internet of things," *IEEE Transactions on*

Industrial Informatics, vol. 19, no. 1, pp. 286–295, 2022.

[78] G. K. Gudur and S. K. Perepu, “Resource-constrained federated learning with heterogeneous labels and models for human activity recognition,” in *International Workshop on Deep Learning for Human Activity Recognition*. Springer, 2021, pp. 57–69.

[79] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, “Blockchain and federated learning for privacy-preserved data sharing in industrial iot,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[80] E. J. De Aguiar, B. S. Faiçal, B. Krishnamachari, and J. Ueyama, “A survey of blockchain-based strategies for healthcare,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–27, 2020.

[81] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[82] M. S. Ons Aouedi, Kandaraj Piamrat, “Hfedsnn: Efficient hierarchical federated learning using spiking neural networks,” in *21st ACM International Symposium on Mobility Management and Wireless Access (MobiWac2023)*. ACM, 2023.

[83] V. Mugunthan, A. Polychroniadou, D. Byrd, and T. H. Balch, “Smpai: Secure multi-party computation for federated learning,” in *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*. MIT Press Cambridge, MA, USA, 2019, pp. 1–9.

[84] M. Fang, X. Cao, J. Jia, and N. Gong, “Local Model Poisoning Attacks to Byzantine-Robust Federated Learning,” in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[85] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning,” in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 1–15.

[86] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, “Reliable federated learning for mobile networks,” *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[87] S. Lu, Y. Zhang, and Y. Wang, “Decentralized Federated Learning for Electronic Health Records,” in *54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2020, pp. 1–5.

[88] I. Hegedűs, G. Danner, and M. Jelasity, “Gossip learning as a decentralized alternative to federated learning,” in *IFIP International Conference on Distributed Applications and Interoperable Systems*. Springer, 2019, pp. 74–90.

[89] J. Passerat-Palmbach, T. Farnan, M. McCoy, J. D. Harris, S. T. Manion, H. L. Flannery, and B. Gleim, “Blockchain-orchestrated machine learning for privacy preserving federated learning in electronic health data,” in *2020 IEEE international conference on blockchain (Blockchain)*. IEEE, 2020, pp. 550–555.

[90] N. Yang, D. Yuan, Y. Zhang, Y. Deng, and W. Bao, “Asynchronous semi-supervised federated learning with provable convergence in edge computing,” *IEEE Network*, vol. 36, no. 5, pp. 136–143, 2022.

[91] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, “Expanding the reach of federated learning by reducing client resource requirements,” *arXiv preprint arXiv:1812.07210*, 2018.

[92] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint arXiv:1610.05492*, 2016.

[93] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, “Fedhealth: A federated transfer learning framework for wearable healthcare,” *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[94] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, “Improving federated learning personalization via model agnostic meta learning,” *arXiv preprint arXiv:1909.12488*, 2019.

[95] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, “A survey of incentive mechanism design for federated learning,” *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 1035–1044, 2021.

[96] P. K. R. Maddikunta, Q.-V. Pham, D. C. Nguyen, T. Huynh-The, O. Aouedi, G. Yenduri, S. Bhattacharya, and T. R. Gadekallu, “Incentive techniques for the internet of things: a survey,” *Journal of Network and Computer Applications*, vol. 206, p. 103464, 2022.

[97] F. Concone, C. Ferdico, G. L. Re, and M. Morana, “A federated learning approach for distributed human activity recognition,” in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2022, pp. 269–274.

[98] A. Augello, G. Falzone, and G. L. Re, “DCFL: Dynamic Clustered Federated Learning under Differential Privacy Settings,” in *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2023, pp. 614–619.

[99] L. Liu, J. Zhang, S. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[100] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, “Clusterfl: a similarity-aware federated learning system for human activity recognition,” in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 54–66.

[101] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[102] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, and Q. Yang, "Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography," *arXiv preprint arXiv:1909.05784*, 2019.

[103] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A Trustworthy Privacy Preserving Framework for Machine Learning in Industrial Iot Systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6092–6102, 2020.

[104] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-Enabled Federated Learning Data Protection Aggregation Scheme With Differential Privacy and Homomorphic Encryption in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4049–4058, 2021.

[105] K. Farooq, H. J. Syed, S. O. Alqahtani, W. Nagmeldin, A. O. Ibrahim, and A. Gani, "Blockchain Federated Learning for In-Home Health Monitoring," *Electronics*, vol. 12, no. 1, p. 136, 2022.

[106] T. Federated, "Tensorflow federated: Machine learning on decentralized data," *Mountain View, CA, USA*, 2019.

[107] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.

[108] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[109] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, and M. Nordlund, "Open-source federated learning frameworks for iot: A comparative review and analysis," *Sensors*, vol. 21, no. 1, p. 167, 2020.

[110] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn *et al.*, "Ibm federated learning: an enterprise framework white paper v0.1," *arXiv preprint arXiv:2007.10987*, 2020.

[111] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 10 320–10 325, 2021.

[112] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[113] X. H. Dun Zeng, Siqi Liang and Z. Xu, "Fedlab: A flexible federated learning framework," *arXiv preprint arXiv:2107.11621*, 2021.

[114] D. Chen, V. J. Tan, Z. Lu, E. Wu, and J. Hu, "Openfed: A comprehensive and versatile open-source federated learning framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5017–5025.

[115] J. O. d. Terrail, S.-S. Ayed, E. Cyffers, F. Grimberg, C. He, R. Loeb, P. Mangold, T. Marchand, O. Marfoq, E. Mushtaq *et al.*, "Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings," *arXiv preprint arXiv:2210.04620*, 2022.

[116] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão *et al.*, "Flower: A friendly federated learning framework," *arXiv preprint arXiv:2007.14390*, 2022.

[117] Q. Zhang, C. Wang, H. Wu, C. Xin, and T. V. Phuong, "Gelu-net: A globally encrypted, locally unencrypted deep neural network for privacy-preserved learning." in *IJCAI*, 2018, pp. 3933–3939.

[118] M. M. Rahman, D. Kundu, S. A. Suha, U. R. Siddiqi, and S. K. Dey, "Hospital patients' length of stay prediction: A federated learning approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 7874–7884, 2022.

[119] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2020.

[120] T. Shaik, X. Tao, N. Higgins, R. Gururajan, Y. Li, X. Zhou, and U. R. Acharya, "Fedstack: Personalized activity monitoring using stacked federated learning," *Knowledge-Based Systems*, p. 109929, 2022.

[121] K. Arikumar, S. B. Prathiba, M. Alazab, T. R. Gadekallu, S. Pandya, J. M. Khan, and R. S. Moorthy, "Fl-pmi: Federated learning-based person movement identification through wearable devices in smart healthcare systems," *Sensors*, vol. 22, no. 4, p. 1377, 2022.

[122] M. J. Baucas, P. Spachos, and K. N. Plataniotis, "Federated learning and blockchain-enabled fog-iot platform for wearables in predictive healthcare," *IEEE Transactions on Computational Social Systems*, 2023.

[123] A. Sarkar, T. Sen, and A. K. Roy, "Grafehty: Graph neural network using federated learning for human activity recognition," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 1124–1129.

[124] A. R. Khan, H. U. Manzoor, F. Ayaz, M. A. Imran, and A. Zoha, "A privacy and energy-aware federated framework for human activity recognition," *Sensors*, vol. 23, no. 23, p. 9339, 2023.

[125] G. Gad and Z. Fadlullah, “Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems,” *Sensors*, vol. 23, no. 1, p. 6, 2022.

[126] D. Cheng, L. Zhang, C. Bu, X. Wang, H. Wu, and A. Song, “Protohar: Prototype guided personalized federated learning for human activity recognition,” *IEEE Journal of Biomedical and Health Informatics*, 2023.

[127] K. Chen, Y. Chai, W. Zhang, S. Wang, J. Shen *et al.*, “Federated unlearning for human activity recognition,” *arXiv preprint arXiv:2404.03659*, 2024.

[128] P. Qi, D. Chiaro, and F. Piccialli, “Fl-fd: Federated learning-based fall detection with multimodal data fusion,” *Information fusion*, vol. 99, p. 101890, 2023.

[129] O. Aouedi, M. A. Bach Tobji, and A. Abraham, “An ensemble of deep auto-encoders for healthcare monitoring,” in *Hybrid Intelligent Systems: 18th International Conference on Hybrid Intelligent Systems (HIS 2018)*. Springer, 2020, pp. 96–105.

[130] A. Halimi, S. Kadhe, A. Rawat, and N. Baracaldo, “Federated unlearning: How to efficiently erase a client in fl?” *arXiv preprint arXiv:2207.05521*, 2022.

[131] L. Tu, X. Ouyang, J. Zhou, Y. He, and G. Xing, “FedDL: Federated Learning via Dynamic Layer Sharing for Human Activity Recognition,” in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys ’21)*, 2021, pp. 15–28.

[132] Q. Shen, H. Feng, R. Song, S. Teso, F. Giunchiglia, H. Xu *et al.*, “Federated multi-task attention for cross-individual human activity recognition,” in *IJCAI*. IJCAI, 2022, pp. 3423–3429.

[133] H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, and X. Cheng, “Fedhar: Semi-supervised online learning for personalized federated human activity recognition,” *IEEE Transactions on Mobile Computing*, 2021.

[134] R. Presotto, G. Civitarese, and C. Bettini, “FedCLAR: Federated Clustering for Personalized Sensor-Based Human Activity Recognition,” in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2022, pp. 227–236.

[135] ——, “Semi-supervised and personalized federated activity recognition based on active learning and label propagation,” *Personal and Ubiquitous Computing*, vol. 26, no. 5, pp. 1281–1298, 2022.

[136] ——, “Federated clustering and semi-supervised learning: a new partnership for personalized human activity recognition,” *Pervasive and Mobile Computing*, vol. 88, p. 101726, 2022.

[137] Y. Li, X. Wang, and L. An, “Hierarchical clustering-based personalized federated learning for robust and fair human activity recognition,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 7, no. 1, pp. 1–38, 2023.

[138] X. Zhou, W. Liang, J. Ma, Z. Yan, I. Kevin, and K. Wang, “2d federated learning for personalized human activity recognition in cyber-physical-social systems,” *IEEE Transactions on Network Science and Engineering*, 2022.

[139] C. Zhang, T. Zhu, H. Wu, and H. Ning, “Perml-fed: enabling personalized multi-level federated learning within heterogenous iot environments for activity recognition,” *Cluster Computing*, pp. 1–16, 2024.

[140] I. Iwan, B. Yahya, and S.-L. Lee, “Federated model contrastive learning with adaptive control variates for human activity recognition,” *Available at SSRN 4810020*, 2024.

[141] S. Ek, F. Portet, P. Lalanda, and G. Vega, “Evaluation of federated learning aggregation algorithms: application to human activity recognition,” in *ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC ’20 Adjunct)*, 2020, pp. 638–643.

[142] G. K. Gudur and S. K. Perepu, “Federated learning with heterogeneous labels and models for mobile activity monitoring,” *arXiv preprint arXiv:2012.02539*, 2020.

[143] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, “Federated meta-learning with fast convergence and efficient communication,” *arXiv preprint arXiv:1802.07876*, 2018.

[144] F.-E. Yang, C.-Y. Wang, and Y.-C. F. Wang, “Efficient model personalization in federated learning via client-specific prompt generation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 159–19 168.

[145] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” *Advances in neural information processing systems*, vol. 30, 2017.

[146] O. Aouedi and K. Piamrat, “Surfs: sustainable intrusion detection with hierarchical federated spiking neural networks,” *ICC 2024*, 2024.

[147] W. Wen, Z. Chen, H. H. Yang, W. Xia, and T. Q. Quek, “Joint scheduling and resource allocation for hierarchical federated edge learning,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5857–5872, 2022.

[148] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[149] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[150] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, “Federated unlearning,” *arXiv preprint arXiv:2012.13891*, 2020.

[151] L. Wu, S. Guo, J. Wang, Z. Hong, J. Zhang, and Y. Ding, “Federated unlearning: Guarantee the right of clients to forget,” *IEEE Network*, vol. 36, no. 5, pp. 129–135, 2022.

[152] J. Dong, L. Wang, Z. Fang, G. Sun, S. Xu, X. Wang, and Q. Zhu, “Federated class-incremental learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 164–10 173.

[153] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, “A survey on intelligent internet of things: Applications, security, privacy, and future directions,” *IEEE Communications Surveys & Tutorials*, 2024.

[154] A. Sacco, F. Esposito, and G. Marchetto, “Restoring application traffic of latency-sensitive networked systems using adversarial autoencoders,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2521–2535, 2022.

[155] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, “Integer quantization for deep learning inference: Principles and empirical evaluation,” *arXiv preprint arXiv:2004.09602*, 2020.

[156] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, “Model pruning enables efficient federated learning on edge devices,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 374–10 386, 2022.

[157] Z. Zhu, J. Hong, and J. Zhou, “Data-free knowledge distillation for heterogeneous federated learning,” in *International conference on machine learning*. PMLR, 2021, pp. 12 878–12 889.

[158] O. Aouedi, “Towards a scalable and energy-efficient framework for industrial cloud-edge-iot continuum,” *IEEE Internet of Things Magazine*, 2024.