A Review of Federated Learning: Algorithms, Frameworks and Applications

Lutho Ntantiso(✉), Antoine Bagula, Olasupo Ajayi, and Ferdinand Kahenga-Ngongo

Department of Computer Science, University of the Western Cape, Cape Town, South Africa
3652045@myuwc.ac.za, ooajayi@uwc.ac.za

Abstract. In today's world, artificial intelligence (AI) and machine learning (ML) are being widely adopted at an exponential rate. A key requirement of AI and ML models is data, which often must be in proximity to these models. However, it is not always possible to "bring data to the model", due to several reasons including legal jurisdictions, or ethical reasons, hence, a "taking the model to the data" might be a viable alternative. This process is called Federate Learning (FL), and it is a ML technique that allows devices or clients to collaboratively learn a shared model from the central server, while keeping the training data local and isolated. This ensures privacy and bandwidth preservation, especially in resource constrained environments. In this paper, a review of FL is done with a view of presenting the aggregation models, frameworks, and application areas, as well as identifying open challenges/gaps for potential research works.

Keywords: Federated Learning · Machine Learning · Federate Averaging

1 Introduction

In recent years data has proven to be one of the most important resources in business operations, decision making, analysis, and research. Data is generated at a fast rate, in different formats, and in large quantities from different sources including IoT (Internet of Things), social and classic media. Due to the diversity of source and type of data. However, data analysis has become more challenging and intensive due to several reasons, including, policies on data privacy and security, networking, and communication constraints, and most importantly because data is often dispersed and exists in the form of isolated islands [1]. Traditional data analysis tools, both statistical and machine learning, primarily work with aggregated data stored in centralized locations. In these classic systems, "data is brought to the analysis models", however this might not be possible in recent contexts due to several reasons. Legislative policies hinder certain types of data (such as sensitive military or governmental information) from leaving the boundaries of the country to which they belong. Similarly, medical related data and personal information that might compromise a person's safety also need to be well guarded. These among other reasons are challenges of traditional data analysis methodologies.

R. A. Saeed et al. (Eds.): AFRICOMM 2022, LNICST 499, pp. 341–357, 2023.
https://doi.org/10.1007/978-3-031-34896-9_20

Federated learning (FL), a recent distributed and decentralized machine learning scheme, has attracted significant attention as a means of mitigating these challenges [2]. In FL clients maintain their data locally, while they "download" a model from the central server for training. In essence, rather than taking the data to the model, the model is taken to the data. Once the model is trained on the client's devices, updates to the model are sent back to the server. The server in turn aggregates these model updates from each of the clients and updates the global model. This process is repeated over multiple training iterations. Figure 1 gives a visual illustration of a FL architecture with a central FL server to which multiple client devices are connected.

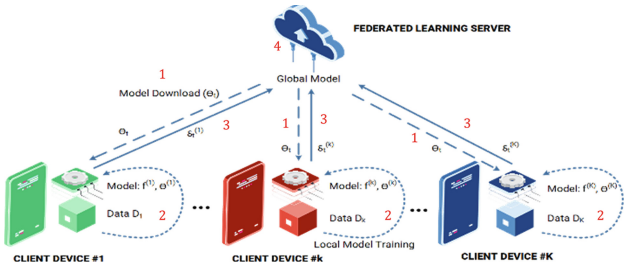

Fig. 1. A Generic Federated Learning Architecture

There are two main factors that make this setting different to distributed training on Graphic Processing Units (GPU), which are communication and heterogeneity. Distributed training on GPU refers to multi-node machine learning algorithms designed to improve performance, increase accuracy, and scale to larger input data sizes [3]. In distributed training, worker nodes work in parallel to speed up the training of ML models [4]. Also, in GPU clusters all clients are on the same local network, which makes communication relatively fast, however in FL devices communicate over the Internet which is comparatively slower. Heterogeneity is presented in two forms, the clients, and the data. Devices and data vary, some devices are faster than others, and some devices will have more data than others, i.e., number of images in mobile phones. We call the variation of these distributions NON IID (independently identically distributed).

The topic centered around data distributions and heterogeneity allowed researchers to introduce different categories of FL which are horizontal federated learning, vertical federated learning, and federated transfer learning. These variations of FL can be applied using different ML techniques on distributed data, across different industries such as healthcare, mobile applications, autonomous vehicles etc. FL is an emerging aspect of AI and has shown significant benefits over traditional ML approaches, including data privacy or security, hardware efficiency, data diversity/heterogeneous data, and real time learning. Unlike other literature that only focus on FL technologies and enabling technologies, this paper aims at presenting a technical overview of FL, specifically the Federated averaging algorithms, in addition to the processes, challenges, advances of FL in general. It views FL from three perspectives, which are required for building sustainable digital infrastructure for cities and rural dwelling, these are: FL aggregation, FL frameworks, and FL application areas.

This paper is structured as follows, Sect. 2 provides a survey of FL in general, while Sect. 3 discusses the various FL aggregation models. Section 4 presents the various state-of-the-art FL frameworks, while various application domains of FL are discussed in Sect. 5. Section 6 presents some open challenges, while Sect. 7 concludes the paper and gives insights into future works.

2 Federated Learning

2.1 Federated Learning

The concept of federated learning has been well researched by numerous authors in literature, however for completeness purposes, this subsection provides a brief review of some of works.

In [5], the authors discussed federated learning through several lenses including, the basic FL fundamentals, FL enabling technologies, the various FL architectures, open challenges including privacy and security, application domains of FL, as well as possible future advancements in FL. Specifically, the authors considered the applications of FL in Artificial intelligence, the Sensor networks, Natural Language Processing, and other contemporary technological domains. In [6], the authors reviewed federated learning and the open problems that exist in federated learning settings. This study also discussed the applications of FL mainly in industrial engineering, six research fronts to address FL for future advancements or optimization. This study [7], addressed the characteristics and challenges of federated learning, as well as an overview of current FL approaches.

2.2 Categorization of Federated Learning

Due to different data distribution patterns, FL can be classified into three different categories: horizontal federated learning, vertical federated learning, and federated transfer learning. In this section we briefly describe these categories.

Horizontal Federated Learning. This FL technique is applied in situations where the datasets share the same feature space, i.e., the datasets have the same sets of features [1]. In 2016 Google proposed a horizontal federated learning framework [8], where a single user using an Android mobile device updates the model parameters locally and uploads these parameters to the cloud, thus collaboratively training the centralized model together with other devices. It is possible that during data transfer, the user's private information may be leaked. Therefore, this framework makes use of secure aggregation [55] and differential privacy measures to ensure privacy during aggregations of user updates. Figure 1 is an illustration of the horizontal FL architecture. In the figure, step 1 depicts the client devices downloading the ML model from the FL server. In step 2 the clients train the downloaded model using localized data and then upload the updated model to the FL server in step 3. The FL server then aggregates the updates from the client devices in step 4.

Vertical Federated Learning. Vertical federated learning or heterogeneous FL can be used in a case where datasets have certain features in common (overlap) [8]. In

essence this framework uses different datasets of different feature space to jointly train a global model [10]. Therefore, with vertical FL the feature dimensions of training data are increased. For example, using the illustration in Fig. 2, if we assume company A to be an ecommerce webstore with data about a customer's book purchase history, and company B to be a review website, such as Bookish, with data about the customer's book reviews. Though both platforms hold different datasets, the datasets have features that overlap to some degree (book title and author). When data about customer 1 is pulled by the central server, more features are available (book title, author, genre, purchase record, read count, and ratings), enabling the central server to better serve the customer, for instance by providing better book recommendations when browsing books on the ecommerce store. Data leakages are bound to occur during data transfer in this setting, hence differential privacy, secure aggregation, and homomorphic encryption are often employed [8].

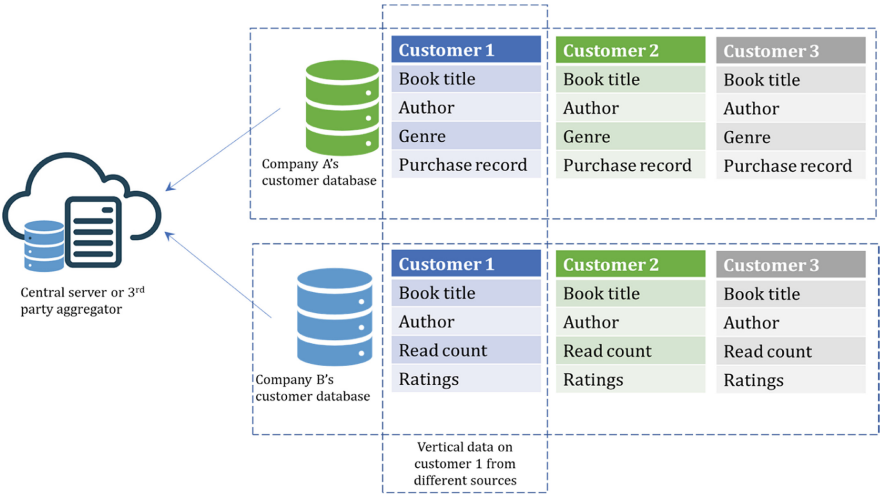

Fig. 2. Illustration of Vertical Federated Learning

Federated Transfer Learning. This applies in cases where two datasets differ not only in sample size but in feature space as well [8]. In essence this technique is useful in settings where the users and user features both rarely overlap. Therefore, to compensate for lack of data or tags transfer learning is used. For example, consider two different companies, one being an ecommerce company in the United States, while the other is a South African bank. Due to geographical restrictions, customers of these institutions have minimal intersection. Moreover, being institutions in different domains, only a small portion (if any) of data features from both parties would overlap. In such cases, transfer learning can be used to carry out effective FL, by optimizing the performance of a task where there is not enough data for training. This technique not only looks at preserving privacy but also solves the issue of small datasets.

3 Federated Aggregation Models

As described in the previous section, most FL architecture consists of a central server which manages the global model and combines updates from the various clients. This section briefly describes the common aggregation models used in the FL environment.

3.1 Aggregation Models

Federated Averaging (FedAvg). It is the very first federated learning algorithm created by Google in 2016 [10] [11] to solve FL problems. FedAvg tries to train a shared model across different clients by minimizing the overall global loss, which is the weighted average of each client's loss. This weighted average is described in (1).

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{1}$$

where $K =$ no. of clients, $F_k(w) =$ client's loss function, $n_k =$ size of client $k's$ dataset.

Each of the losses is weighted by the size of the client $k's$ dataset. Therefore, devices with larger datasets will have larger weighted losses. FedAvg is not the most efficient FL algorithm due to the simplified assumptions that the model makes, which are:

- All sample devices will complete E epochs of local stochastic gradient descent. It is important to note that some devices take longer than others, which can negatively impact the speed of convergence.
- Convergence is not guaranteed if the data is highly heterogeneous.
- Devices are weighed by the proportion of the data that they have. This could mean that the algorithm might favor clients with more data.

Other variants of FedAvg have been developed to address some of the limitations of the original version.

FedProx. The authors in [12] proposed FedProx, which is a technique that allows devices to do variable amounts of work. This approach introduces a regularization term or proximal term that penalizes large changes in weights and helps with convergence on heterogeneous data. The authors reported that FedProx can achieve an average of 22% absolute accuracy improvement over the base FedAvg method. FedProx is defined in (2)

$$h_k\left(w; w^t\right) = F_k(w) + \frac{\mu}{2}\left|\left|w - w^t\right|\right|^2 \tag{2}$$

where, $\frac{\mu}{2}\left|\left|w - w^t\right|\right|^2$ is a Proximal term, $\frac{\mu}{2}$ is the hyper-parameter that controls the penalized amount.

Q-FedAvg. With heterogeneous data, model performance varies since different distributions may require different sets of features. Recent research study introduced two ways of addressing this. Ref. [13] proposed an algorithm called Q-FedAvg. This technique encourages a fairer or uniform accuracy distribution across devices in federated networks, which means the model performs similarly across all devices. Q-FedAvg

is a lightweight and scalable distributed method used to solve Q-FFL (Fair Federated Learning) in massive, federated networks. Q-FFL is a method that introduces fairness in federated settings, and Q-FedAvg is a technique that improves the efficiency of Q-FFL. Rather than weighting devices on the proportion of the data they have, Q-FedAvg penalizes the worst performing devices, incentivizing the model to improve performance on these devices [13].

$$f_q(w) = \sum_{k=1}^{m} \frac{p_k}{q+1} F_k^{q+1}(w) \tag{3}$$

Per-FedAvg. The base FedAvg model does not cater for different data distributions, such as in heterogeneous settings (non-IID), hence the resulting global model obtained by minimizing the average loss could perform poorly when applied to the local datasets of each user. The authors in [14] thus focused on data in heterogeneous settings, where the probability distributions of devices are not identical (i.e., non-IID). They overcome the challenge by incorporating personalization. This technique builds on the Model-Agnostic Meta Learning approach introduced in [15], that formulates FL as a multi-task problem, where each client's distribution is a separate task. The was to find an initial shared model that current or new clients can easily adapt to their local dataset by running one or a few sets of gradient descent.

$$f(w) = \sum\nolimits_{k=1}^{m} P_k F_k(w) \tag{4}$$

Per-FedAvg is described in (4), where the loss function is changed from the loss on weights from the current round to the weights after one step of gradient descent as shown in (5).

$$f(w) = \sum\nolimits_{k=1}^{m} P_k F_k((w- \propto \nabla F_k(w))) \tag{5}$$

where $((w- \propto \nabla F_k(w)))$ is weights after one step of gradient descent.

Table 1 summarizes the averaging various models described thus far.

3.2 Evaluation Dataset

This subsection presents some of the datasets used for testing and evaluating the federated aggregation models discussed in the previous section. Table 2 gives a summary of models and the corresponding datasets utilized.

MNIST [18] and Sharespare [20] were the most utilized datasets, while all models except Per-FedAvg curated some form of synthetic dataset for their testing.

Table 1. FL Algorithm Summary

Ref.	FL Method	Description	Algorithm	Remark
[10, 11]	FedAvg	First federated learning algorithm that tries to train a shared model across different clients.	$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w)$	Does not take heterogeneity of devices into consideration. Does not guarantee convergence.
[12]	FedProx	An improvement over FedAvg to support difference in client workloads. Introduces a regularization term that penalizes large changes in weights.	$\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2$	Helps with convergence on heterogeneous data
[13]	Q-FedAvg	A lightweight, scalable, fair, and distributed FL model.	$\min_w f_q(w) = \sum_{k=1}^{m} \frac{p_k}{q+1} F_k^{q+1}(w)$	Focuses on fairness in FL
[14]	Per-FedAvg	This technique formulates FL as a multi-task problem where each client's distribution is a separate task.	$\min_w f(w) = \sum_{k=1}^{m} p_k F_k(w)$ $\min_w f(w) = \sum_{k=1}^{m} p_k F_k\left((w - \propto \nabla F_k(w))\right).$	Builds on Model-Agnostic Meta learning Approach

Table 2. Evaluation datasets

Model	CIFAR-10 [15]	EMNIST [16]	Fashion MNIST [17]	MNIST [18]	Sent140 [19]	Shakespare [20]	Synthetic
FedAvg	No	No	No	Yes	No	Yes	Yes
FedProx	No	Yes	No	Yes	Yes	Yes	Yes
Q-FedAvg	No	No	Yes	No	Yes	Yes	Yes
Per-FedAvg	Yes	No	No	Yes	No	No	No

4 Federated Learning Frameworks

FL is a relatively new concept in terms of distributed ML; hence the development of efficient FL models would require the selection of appropriate frameworks for their implementation. This section presents some of the common frameworks used for FL.

FATE [21]. FATE (Federated AI Technology Enabler) was developed in 2019 by Webank. The objective of FATE was to support a collaborative and distributed AI ecosystem with cross-silo data applications while meeting compliance and security requirements. It has several components, including FATEFlow – an FL management pipeline,

FederatedML - ML library, FATEBoard –visualization tool, FATE Serving – high performance computing platform for FL models, Federated Network - communication, and KubeFATE – cloud-based FL workload manager.

Flower [22]. Flower is an open-source FL framework, under Apache 2.0 License and adopted by major research organizations in both academia and industry. It is a unified approach to federated learning, analytics, and evaluation with support for multiple ML frameworks (PyTorch, Keras, TensorFlow), different operating systems (Android, IOS, Windows) and platforms (mobile, desktop, and Cloud).

Substra [23]. Substra is a FL framework that places emphasis on privacy and traceability in distributed ML. It employs a form of distributed ledger technology for data privacy and traceability. Like Flower, Substra also supports multiple programming languages and algorithms.

OpenFL [24]. Open Federated Learning (OpenFL) is a Python 3 library designed for secure and collaborative FL. It is made up of collaborators – which train ML models using localized datasets, aggregators – which combine updates from collaborators into a global model, a director – which coordinates the federation process, and envoy - which manage the execution of workloads on the collaborators.

TensorFlow Federated [25]. TensorFlow Federated (TFF) is an open-source framework for FL on decentralized data developed by Google. The main motivation behind TFF was Google's need to implement mobile keyboard predictions and on-device search. TFF has two major components, viz. Federated Core API – which combines TensorFlow with distributed communication for implementing distributed computations) and Federated Learning API - a high-level API that enables existing ML models to be plugged into TFF.

Other FL frameworks include IBM Federated Learning [26] and NVIDIA Clara [27].

5 Applications of Federated Learning

To our knowledge, the Internet of things (IoT), healthcare, NLP, and transportation are some of the key requirements for sustainable city and rural dwellings. Globally, FL has been applied to integrate existing solutions/systems in these areas. This section presents some documented applications of FL in the aforementioned domains.

5.1 Federated Learning in Internet of Things (IoT)

The Internet of Things has penetrated a broad range of aspects in modern life, leading to the emergence of diverse intelligent systems, smart devices, and applications [28]. With the plethora of vastly distributed and connected smart devices, we can acquire insights on massive data. These data insights can be leveraged to train advanced AI models that serve in the integration and development of smart devices to help society in all aspects [29]. To benefit from this data, one approach is to aggregate these distributed data onto centralized server(s) for analysis and modeling. However, this approach can be ineffective as data transmission may impose privacy risks due to data leakage. An alternative

approach is FL, a novel concept that can address the privacy and security concerns data collection and aggregation [29]. Nevertheless, in complex IoT environments there are still challenges that FL models must face. These include device heterogeneity (different storage, computational power, communication range), data heterogeneity (non-IID data), model heterogeneity, expensive communication, and privacy concerns. To tackle some of these challenges, [28] proposed personalization on devices, data, and models. Personalized FL has attracted significant attention as it aims to mitigate heterogeneity and achieve a quality personalized model on devices in the network.

Below we briefly discuss some of the most significant challenges of FL models in complex IoT environments, specifically device and data heterogeneity.

Device Heterogeneity. Refers to the large number of IoT devices that differ in hardware, network protocols, communication capacity, computing, and storage. In complex IoT settings the main constraint with FL is communication, because devices in the network are relatively slow or frequently offline. In some scenarios devices with limited computing capacity may become stragglers as they take long to report model updates, while others may drop out of the network due to poor connectivity causing a negative effect on the learning process [28].

Data Heterogeneity. Refers to non-IID data distributions of devices found in a FL environment. Statistically, user data can be non-IID in various forms, such as skew feature distribution, skew label distribution, and concept shift [28]. In most FL settings (e.g., medical data across hospitals, financial data across financial institutions) devices generate and collect data in a non-IID distributed manner leading to statistical shift among them [30].

The current research work is exploring different ways of mitigating the above-mentioned problems and other arising issues of FL in IoT environments. As stated previously, [28] proposed an edge computing framework for personalized FL, which helps mitigate heterogeneity in IoT applications. Edge computing brings technologies that allow computation and data storage at the network edge so that computing can be carried out closer to devices or data sources. Edge computing can address concerns such as latency, limited battery life in mobile devices, bandwidth costs, security, and privacy [31]. By bringing computing closer to the edge, the issue of device heterogeneity (varied computation and communication) can be addressed, as each device on the network can offload its intensive computation learning task to the edge that provides fast processing and low latency. Furthermore, statistical and model heterogeneity of data can also be addressed with Edge computing, as the edge nodes can participate in the collaborative training of global models under the coordination of a central cloud server [28].

5.2 Healthcare

FL has been applied in various sub-domains in healthcare, including in the diagnosis of diseases, cardiac related cases, and drug related complications. These three use cases are discussed below.

Diagnosis of COVID-19. The healthcare industry is known to be one of the industries that carry sensitive information. Therefore, due to several challenges associated with

traditional ML and cloud-based infrastructures such as security and privacy during data transmission, edge computing has proven to be an efficient computing resource for storing and processing large volumes of medical data, as it brings high quality computing resources closer to the clients (hospitals).

In [32] the authors leverage edge-computing capabilities in the healthcare industry by analyzing and evaluating the capabilities of AI processing on isolated clinical data related to COVID-19. The work proposed a model for automatic diagnosis of COVID-19 using the clustered federated learning (CFL). Two datasets were utilized, the first containing chest X-ray images, while the other contained chest ultrasound images, upon which binary classification was carried out to differentiate between COVID-19 chest images and normal chest images. VGG16 convolutional neural network (CNN) model was used for training, while focal loss was used to address data imbalance. The proposed CFL model was benchmarked against a specialized FL model and a multimodal CFL. Obtained results showed a comparable performance between the proposed framework and the specialized FL models. The multimodal variant was susceptible to overfitting, hence had to be stopped as certain point.

Hospitalizations Due to Cardiac Events. In [33] the authors proposed a FL method to predict hospitalizations due to cardiac events. An iterative cluster primal dual splitting (cPDS) algorithm was built to solve a sparse Support Vector Machine (SVM) problem in a decentralized manner. The work sought to address three challenges tied to healthcare data, which are distributed data residing in isolated locations (hospitals, smart devices), data aggregation in a single database, which is infeasible due to data size and privacy concerns, and the development of a scalable framework to leverage the growing data. Data for the study was extracted from the EHR system and contained demographic data, medical history, and drug prescription history from 2001 to 2012 and containing at least one heart diagnosis between 2005 and 2010. The proposed framework was tested based on its ability to predict hospitalization due to cardiac conditions within a calendar, using data from previous years. Results from the experiment showed that the cPDS algorithm converged faster with less communication overhead compared to some of the distributed algorithms in this experiment and the accuracies of all compared algorithms were relatively equal.

Predicting Adverse Drug Reaction (ADR) and Mortality Rate. Ref. [34] proposed a model for predicting Adverse Drug Reaction (ADR) and Mortality rate using FL as well as demonstrating the effectiveness of the FL when privacy preserving measures are in place. Two use cases were considered – prediction of ADR and modeling in-hospital patient mortality, for which two datasets containing over a million patient records were used. For the first, the Limited Market Scan Explores Claims-EMR Data (LCED) dataset was used, while the Medical Information Mart for Intensive Care (MIMIC) dataset was used for the second use case. Three ML algorithms (perceptron, Logistic Regression, and SVM) were compared in a FL set up across 10 sites. The proposed distributed FL model was compared to a centralized learning setting, and obtained results revealed that the FL model achieved comparable performance to centralized learning for both tasks. It was also inferred that though the privacy in the FL setting was guaranteed the security model adopted affected the predictive capability of the FL model.

5.3 Natural Language Processing (NLP)

NLP helps understand human language in a better way [35]. However, this field of study requires massive amounts of data to accurately train models. Like with the previous application areas, the data collection process comes with issues related to hardware, security, privacy, and communication. In this section we go through a few use cases of NLP in FL.

Institutional Federated Learning for NLP. Ref. [36] applied FL to the TextCNN NLP model, to classify the intents within sentences. They also applied differential privacy to protect the clients during the training process. This experiment made use of the TREC dataset, a publicly available dataset for NLP text applications, which contains about 5952 data points. The FL process was done with the coMind ML framework [37], which supports distributed training on GPUs with a federated average optimizer [36]. The experiment was conducted on a simulated FL setup within a local area network, with one central server and 4 client devices. RSA encryption was used as the security measure for the communication between devices and the central server, while Tensorflow was used to develop the TextCNN model. Results of experimental simulations revealed the following: i.) an accuracy of 91.2% for the centralized model; ii.) a 4% drop in accuracy when the FL model with distributed data was used; iii.) the FL model was sensitive to data load amongst clients, hence imbalance in data across the clients resulted in up to 10% dip in performance; iv.) differential privacy improved security of the model through the introduction of noise to the dataset.

Federated Learning for Mobile Keyboard Prediction. There are multiple constraints when it comes to the development of mobile keyboard models. For instance, to run these models on low and high-end devices, these models need to be small with low inference time [38]. User experience is key when it comes to such settings, as users would expect a visible keyboard response within a very small amount of time. With the current frequency with which mobile keyboards are used, device power could quickly deplete if CPU usage was not constrained [38]. As a result, these models are sometimes limited to a certain data size. To perform next word prediction in virtual keyboards of smart mobile phones (and tablets), the authors trained a recurrent neural network language model using a FL model. The work presented in the work showed the benefit of training models on the client's devices without the transmission of sensitive user data to servers. For the experiment, the authors compared two forms of training - server-based training using stochastic gradient descent (SGD) and training on devices using the FedAvg algorithm [11]. For the server-based training, data containing about 7.5 billion English language sentences collected from Google keyboard (GBoard) in the United States was used. For the FL model, data from local caches of the client's Google keyboard was used. FedAvg was then used to aggregate the client SGD updates. Recall was used as a performance metric and obtained results showed that the FL model gives better recall values compared to server trained model.

Secure and Efficient FL Framework for NLP. FL has provided various ways of mitigating some of the challenges in centralized ML settings, however some of these solutions either require expensive communication, a trusted aggregator, or heavyweight

cryptographic protocols. In [39] a secure and efficient FL framework (SEFL) was introduced that disposes of the need for trusted entities, obtains comparable or better accuracy compared to existing FL frameworks, yet is flexible to client dropouts. SEFL makes use of two non-colluding servers, the Aggregation Server (AS) that collects the encrypted client updates and securely aggregates them, and a Cryptographic Service Provider (CSP) that manages the decryption key. In this design the CSP generates key pairs (secret key and public key) and stores the secret key locally but publishes the public key to all clients. The AS performs encryption on encrypted local updates. Therefore, to decrypt the aggregated updates, these two entities need to collaborate, since the CSP is the only entity that has control over the decryption key. Additional operations of encryption and decryption in FL settings could lead to high computation and communication overhead. To minimize the number of required cryptographic operations during training, the NLP model is trained with reduced volumes of local updates and weight storage. This experiment also made use of a crypto friendly Block-Hankel matrix-based pruning, which aims to achieve symmetric and balanced downloading and uploading communication. The proposed framework was evaluated by conducting experiments using Long Short-term Memory (LSTM) [40] and Transformer model [41] on the WikiText-2 dataset [42]. Results show that the SEFL framework can produce accurate models even when there are several client dropouts in the network.

5.4 Transportation

With the increase of IoT sensors in vehicular networks, it is important to collect data and train ML models that can be used for vehicle and traffic management. Since the data in such scenarios are distributed across the various vehicles, FL can be applied. One particular use case of interest is in predicting energy demand of electric vehicles (EVs). EVs have become one of the most sustainable solutions in transportation, as they help reduce oil consumption, high energy usage and gas emissions. Ref. [43] proposed some ML approaches that helped improve the efficiency and accuracy of energy demand prediction as well as reduce communication overhead in EV networks. Firstly, the researchers introduced a central based approach in which a charging power station (server) collected data from all charging stations (clients) close by and performed energy demand learning to predict energy demand using the data from the clients. This approach was deemed to be inefficient because it required data sharing between the server and distributed clients which resulted in communication overheads, privacy concerns for the clients and the EVs that utilized them. Therefore, federated energy demand learning (EDL) was introduced to address the above issue. To further improve accuracy on the proposed approach, this research introduced a clustering-based EDL, which also helps reduce dimensionality on dataset [44], minimizing biased prediction [45]. The authors compared their proposed methods to conventional machine learning models which are, decision trees, random forest, support vector regressor, k-neighbors regressor, stochastic gradient descent and multilayer perceptron, using a dataset obtained from charging stations across Dundee city between the period of 2017 and 2018. This data set has 65601 transactions, which include the charging station ID of 58 charging stations, transaction ID for each charging station, electric vehicle charging time and consumed energy for each transaction. Experimental results obtained from comparing the conventional ML models and the proposed

frameworks on various training and testing set ratios, revealed that the proposed models improved accuracy of energy demand prediction (EDP) by 24.63% and decreased communication overhead by 83.4% compared to the other models.

6 Open Challenges in Federated Learning

There are several challenges in FL settings, including systems heterogeneity, statistical data heterogeneity, expensive communication, and privacy concerns; most of which have been discussed earlier During information sharing in such setups, we would encounter the previously stated problems which may result in devices dropping out of the network. Since many of these problems are inherently interdisciplinary, solving them requires techniques from distributed optimization, cryptography, security, differential privacy, fairness, compressed sensing, information theory and more [46]. The aim of this section is to highlight FL challenges, specifically privacy, security, and fairness.

There are two main techniques to preserve privacy introduced by [47, 48] which are differential privacy and secure aggregation. Differential privacy is defined in terms of the application-specific concept as is the case with end-to-end databases [47]. There are many properties that make differential privacy convenient or functional to certain settings such as composability, group privacy, and robustness to information [47]. In federated settings, this technique improves privacy by studiously adding noise on the datasets or outputs. This can either be done at client level, server level or at both (hybrid approach). However, in practice adding noise at client level is not practical since each device/client may only have little data.

Secure aggregation is a cryptographic technique wherein a group of clients each hold a private value and collaborate to compute an aggregate value, without revealing to one another any information regarding the private value except what is learnable from the aggregate value [48]. In essence this technique ensures privacy on individual model updates is preserved. Another technique of preserving privacy is to add randomness in the process of device checking in [49], or by shuffling the model updates sent by devices to the central [50]. It is imperative to note that privacy is an important domain of FL, therefore whenever we build any FL model, we need to ensure that it aligns well with differential privacy and secure aggregation. However, this also introduces another challenge of ensuring fairness in the FL process. Balancing privacy/security concerns with fairness is an open challenge in FL.

One of the most important aspects of FL is fairness. Several research efforts have been made to address this challenge, such as in [51], where the authors attempted to tune the overall loss function to improve the model to perform equally on all device types. The concept of fairness can be viewed from multiple dimensions. For instance, in [2] it was argued that devices that contribute more to a FL network should be better rewarded. Similarly, [52] proposed a collaborative fair FL framework which aimed to achieve collaborative fairness by evaluating the contributions of clients and iteratively updating their respective reputations. However, most FL frameworks simply ignore the aspect of device contribution. From the research landscape on the concept of fairness, it seems to be relative. Some researchers consider it more of a policy related question than a technological question, while others consider it technological. Ultimately, it narrows down to how devices are viewed and weighed in a system or network.

The techniques used to build and interpret models need to adapt to work on heterogeneous data. In FL settings, data cannot be assumed to be independently and identically distributed, as clients can have different data distributions which can affect the speed of convergence. The authors in [53] analyzed the asymptotic convergence of these algorithms under different data distributions. Some researchers argue that even-out the distributions of data across clients can help tackle the issue of heterogeneity by ensuring that clients have relatively equal distribution of data. However, this introduces two new challenges, i.) How can data on globally distributed clients be controlled? This is synonymous to wanting to ensure that all data on people's smartphones are of equal sizes globally. ii.) ensuring equality of data across client nodes, might involve redistributing data by moving data from clients with more data to those with less data. Doing this introduces security concerns, and circles back to the privacy and security challenges discussed above. In [54] a technique called Federated Augmentation was proposed, where each device collectively trains a generative model, and thereby augments its local data towards yielding an IID dataset.

Table 3 summaries the various aspects of FL discussed, presenting them as a form of FL taxonomy.

Table 3. Federated learning taxonomy

Categorization	Method	Description	Application
Data Structure	Horizontal Federated Learning	Data with overlapping feature sets	Regression techniques
	Vertical Federated Learning	Data with different feature space, hence increased feature dimension	Neural Networks and Deep Neural Networks
	Federated Transfer Learning	Increased feature size and sample	Transfer Learning
Privacy techniques	Differential Privacy	Ensures privacy by studiously adding noise on the datasets or outputs	Traditional ML, Neural Networks
	Secure Aggregation	Clients hold a private value and collaborate to compute an aggregate value, without revealing information to each other	Federated learning, Regression
Heterogeneity techniques	Edge Computing	Mitigates heterogeneity by bringing technologies that allow computation and data storage near the network edge	Device heterogeneity, statistical data heterogeneity, model heterogeneity
	Asynchronous Computing	Solves for high-cost communication and latency	Device Heterogeneity

7 Conclusion

The growing demand for federated learning (FL) technology led to the development of tools and frameworks that can be used to handle vastly distributed data. The articles reviewed in this work show that FL cuts across various domains, including machine learning, information theory, statistics, fairness, privacy, and security; and has been successfully applied in various fields including, healthcare, transportation, and the Internet of Things. However, despite the advances, there remain several open challenges. This paper gave insights on ways on how techniques applied in this field can be used to improve efficiency, security and ensure fairness in FL networks. Some potential research gaps identified are related to fairness and privacy in complex FL networks. Exploration of these areas might be an avenue for future research work.

References

1. Yang, Q., Yang, L., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol **10**(2), 1–19 (2019)
2. Zhang, J., Li, C., Robles-Kelly, A., Kankanhalli, M.: Hierarchically fair federated learning. pp. 1–16 (2020)
3. Galakatos, A., Crotty, A., Kraska, T.: Distributed machine learning. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems. Springer, New York (2018). https://doi.org/10.1007/978-1-4614-8265-9_80647
4. Baccam, N., Gilley, S., Coulter, D., Martens, J.: Distributed training with azure machine learning," microsoft [Online]. https://docs.microsoft.com/en-us/azure/machine-learning/concept-distributed-training. Accessed 20 Aug 2021
5. Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., Jararweh, Y.: Federated learning review: fundamentals, enabling technologies, and future applications. Inf. Process. Manage. **59**(6), 103061 (2022)
6. Li, L., Fan, Y., Tse, M., Lin, K.: A review of applications in federated learning. Comput. Ind. Eng. **149**(2020), 106854 (2020)
7. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. IEEE Signal Process. Mag. **37**(3), 50–60 (2020). https://doi.org/10.1109/msp.2020.2975749
8. Gao, Y., Li, W., Yu, B., Bai, H., Xie, Y., Zhang, C.: A survey on federated learning. Knowl. Based Syst. **216** (2021)
9. Gooday, A.: Federated learning types: Understanding the types of Federated Learning. OpenMind. [Online]. https://blog.openmined.org/federated-learning-types/
10. Kelvin. Introduction to Federated Learning and Challenges. Towards Data Science (2020). [Online]. https://towardsdatascience.com/introduction-to-federated-learning-and-challenges-ea7e02f260ca
11. Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., Aguera, B.: Communication-efficient learning of deep networks from decentralized data. International Conference on Artificial Intelligence and Statistics (AISTATS), Florida, 2017
12. Li, T., Sahu, A., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: MSysConference 2020 (2020)
13. Li, T., Sanjabi, M., Beirami, A., Smith, V.: Fair resource allocation in federated learning. arXiv preprint arXiv:1905.10497 (2019)

14. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. In: Conference on Neural Information Processing Systems (NeurIPS 2020) (2020)
15. V. Smith, C. Chiang, M. Sanjabi and A. Talwalkar, "Federated Multi-Task Learning," Conf. on Neural Information Processing Systems, California, 2017
16. [EMNST] Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: an extension of MNIST to handwritten letters. arXiv preprint arXiv:1702.05373 (2017)
17. [FashionMNST] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
18. [MNST] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradientbased learning applied to document recognition. In: Proceedings of the IEEE (1998)
19. [sent140] Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009)
20. [Shakespare] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.Y.: Communication-efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics (2017)
21. Liu, Y., Fan, T., Chen, T., Xu, Q., Yang, Q.: FATE: an industrial grade platform for collaborative learning with data protection. J. Mach. Learn. Res. **22**(226), 1–6 (2021)
22. Beutel, D., Topal, T., Mathur, A., Qiu, X., et al.: Flower: a friendly federated learning research framework. arXiv preprint arXiv:2007.14390 (2020)
23. Galtier, M., Marini, C.: Substra: a framework for privacy-preserving, traceable and collaborative machine learning. arXiv preprint arXiv:1910.11567 (2019)
24. Reina, G., Gruzdev, A., Foley, P., Perepelkina, O., et al.: OpenFL: an open-source framework for federated learning. arXiv:2105.06413 (2021)
25. TensorFlow. TensorFlow Federated: Machine Learning on Decentralized Data. [Online]. https://www.tensorflow.org/federated. Accessed 1 Aug 2022
26. Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., et al.: IBM federated learning: an enterprise framework white paper v0. 1. arXiv preprint arXiv:2007.10987.(2020)
27. Wen, Y., Li, W., Roth, H., Dogra, P.: Federated Learning powered by NVIDIA Clara [Online]. https://developer.nvidia.com/blog/federated-learning-clara/. Accessed 1 Aug 2022
28. Wu, Q., He, K., Chen, X.: Personalized federated learning for intelligent IoT applications: a cloud-edge based framework. In: IEEE Computer Graphics and Applications (2020)
29. Jiang, J., Kantarci, B., Oktug, S., Soyata, T.: Federated learning in smart city sensing: challenges and opportunities. Sensors **20** (2020)
30. Li, Y., Zhou, W., Wang, H., Mi, H., T.: Hospedales, FedH2L: Federated learning with model and statistical heterogeneity (2021)
31. Shi, W., Dustdar, S.: the promise of edge computing. Computer **49**(5), 78–81 (2016)
32. Qayyum, A., Ahmad, K., Ahsan, M., Al-Fuqaha, A.: Collaborative federated learning for healthcare: multi-modal COVID-19 diagnosis at the edge. J. Open Comput. Soc. **3**, 172–184 (2021)
33. Brisimia, S., Chena, R., Melac, T., Olshevskya, A., Paschalidis, C.: Federated learning of predictive models from federated Electronic Health Records. Int. J. Med. Inform. **112**, 59–67 (2018)
34. Choudhury, O., Gkoulalas-Divanis, A., Salonidis, T., Sylla, I., et al.: Differential privacy-enabled federated learning for sensitive health data. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver (2019)
35. Mammen, P.: Federated learning: opportunities and challenges. In: Association for Computing Machinery, Washington (2021)
36. X. Zhu, J. Wang, Z. Hong, and J. Xiao, "Empirical Studies of Institutional Federated Learning for Natural Language Processing," Association for Computational Linguistics, pp. 625–634, 2020

37. Roman, A.: coMind collaborative machine learning framework (2019)
38. Hard, A., Rao, K., Mathews, R., Ramaswamy, S., et al.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)
39. Wang, C., Deng, J., Meng, X., Wang, Y., et al.: A Secure and efficient federated learning framework for NLP. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana (2021)
40. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al.: Attention is all you need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), California (2017)
42. Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models. In: ICLR, California (2017)
43. Saputra, Y., Hoang, D., Nguyen, D., Dutkiewicz, E., et al.: Energy demand prediction with federated learning for electric vehicle networks. In: IEEE Global Communications Conference (GLOBECOM2019), Waikoloa, HI, USA (2019)
44. Hea, Y., Kockelman, K., Perrine, K.: Optimal locations of U.S. fast charging stations for long-distance trip completion by battery electric vehicles. J. Clean. Prod. **214**, 452–461 (2019)
45. Li, W., Logenthiran, T., Phan, V., Woo, W.: Implemented IoT-based self-learning home management system (SHMS) for Singapore. IEEE Internet of Things J. **5**(3), 2212–2219 (2018)
46. Kairouz, P., Brendan McMahan, H., Avent, B., Bellet, A., et al.: Advances and open problems in federated learning. Found. Trends Mach. Learn. **4**(1), (2021)
47. Abadi, M., Chu, A., Goodfellow, I., Brendan McMahan, H., et al.: Deep learning with differential privacy. In: ACM Conference on Computer and Communications Security, Vienna (2016)
48. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., et al.: Practical secure aggregation for federated learning on user-held data. In: International Conference on Neural Info. Processing Systems (NIPS), Barcelona (2016)
49. Balle, B., Kairouz, P., Brendan McMahan, H., Thakkar, O., Thakurta, A.: Privacy amplification via random check-ins. Neural Info. Process. Syst. **33,** 4623–4634 (2020)
50. Erlingsson, U., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: from local to central differential privacy via anonymity. In: ACM-SIAM Symposium on Discrete Algorithms (SODA) (2020)
51. Mohri, M., Sivek, G., Suresh. A.T.: Agnostic federated learning. In: International Conference on Machine Learning, PMLR 2019, pp. 4615–4625 (2019)
52. Lyu, L., Xu, X., Wang, Q., Yu, H.: Collaborative fairness in federated learning. In: Yang, Q., Fan, L., Yu, H. (eds.) Federated Learning. LNCS, vol. 12500, pp. 189-204. Springer, Cham (2020).https://doi.org/10.1007/978-3-030-63076-8_14
53. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of FedAvg on Non-IID. In: ICLR (2020)
54. Jeong, E., Oh, S., Kim, H., Park, J., et al.: Communication-efficient on-device machine learning: federated distillation and augmentation under non-IID private data. In: Neural Info. Processing Systems (NIPS), Montreal (2018)
55. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)