Federated Learning in a Medical Context: A Systematic Literature Review

BJARNE PFITZNER, NICO STECKHAN, and BERT ARNRICH, University of Potsdam, Germany

Data privacy is a very important issue. Especially in fields like medicine, it is paramount to abide by the existing privacy regulations to preserve patients' anonymity. On the other hand, data is required for research and training machine learning models that could help gain insight into complex correlations or personalised treatments that may otherwise stay undiscovered. Those models generally scale with the amount of data available, but the current situation often prohibits building large databases across sites. So it would be beneficial to be able to combine similar or related data from different sites all over the world while still preserving data privacy. Federated learning has been proposed as a solution for this, because it relies on the sharing of machine learning models, instead of the raw data itself. That means private data never leaves the site or device it was collected on. Federated learning is an emerging research area and many domains have been identified for the application of those methods. This systematic literature review provides an extensive look at the concept of and research into federated learning and its applicability for confidential healthcare datasets.

CCS Concepts: • **Security and privacy** → **Privacy-preserving protocols**; • **Computing methodologies** → *Machine learning*; **Distributed artificial intelligence**; • **Applied computing** → **Life and medical sciences**; • **General and reference** → *Surveys and overviews*.

Additional Key Words and Phrases: federated learning

ACM Reference Format:
Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich. 2020. Federated Learning in a Medical Context: A Systematic Literature Review. *ACM Trans. Internet Technol.* 1, 1, Article 1 (January 2020), 37 pages. https://doi.org/10.1145/3412357

1 INTRODUCTION

In a data-centred world, where people are expected to share their data willingly to use services, it is very important to preserve data privacy in areas that are very sensitive. This may include financial information, personal images or medical records. In medicine, for example, doctors struggle with setting up multi-centre studies, because they have to deal with how and where to store the collected patient data, write ethics proposals and wait for lengthy confirmation periods thereof. Article 5 of the European General Data Protection Regulation (GDPR) defines the concepts of *data minimisation*, meaning only relevant data for a study can be collected, and *purpose limitation*, meaning that even after the ethics proposal is approved, the data can only be used for the purpose it was collected for, any future research is restricted and requires an ethics amendment and consent of patients. Also, personalised medicine approaches, which try to adapt treatment specifically to individual patients, could benefit from a way of clustering similar patients and making more informed guesses for the patients' needs. So ideally sensitive existing databases can be used for different directions of research, without the possibility of privacy violations. One way to do this is pseudonymisation or de-identification, where certain identifiable parts of data, such as name, address or social security number are replaced by a pseudonym to preserve a person's privacy. This strategy is not completely secure, and there have been cases where pseudonymised

Authors' address: Bjarne Pfitzner, bjarne.pfitzner@hpi.de; Nico Steckhan, nico.steckhan@hpi.de; Bert Arnrich, bert.arnrich@hpi.de, Digital Health Center, Hasso Plattner Institute, University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, Potsdam, 14482, Germany.

Manuscript submitted to ACM

data could be traced back to individuals [94, 117]. Emerging from those problems, and as a means to distribute the computational load of training a machine learning (ML) model, federated learning (FL) was proposed.

This systematic literature review is aimed at providing a deep dive into the topic of FL and its development. The focus, especially in Sections 3.6 and 4 is laid on the usability of FL for healthcare and health-related data and we will often refer to this throughout the paper.

The term FL was first used by McMahan et al. [47] and describes a distributed and privacy-preserving way of training an ML model without others accessing private data. Instead of sharing the data directly amongst non-trusting parties, FL relies on sharing model parameters that can be aggregated to a joint model. An FL system follows a client-server architecture with one server, who is responsible for facilitating the training, building the model and making it available to all clients, who are training the model on their local datasets. This novel idea stands in contrast to similar, previously known ML types for federated datasets. Distributed ML assumes a centralised dataset which can be distributed to several worker machines in the best way possible [114], and Machine Learning as a Service (MLaaS) describes a system in which a provider hosts an ML model, and clients can upload their data to receive a classification for it and promote model training [115].

FL, on the other hand, is used for situations in which data is:

- **Massively distributed:** A large number of clients (up to millions) which might be scattered all over the world hold relevant data. Although FL systems between hospitals would probably not be as massively distributed, one can imagine using sensor data collected on smartphones for medical purposes, which would require dealing with a large number of clients.
- **Non-IID:** Data collected by different participating clients originate from different distributions, and is thus not independent and identically distributed (IID). Hospitals, for example, see patients from widely different demographics, so it is unfeasible to assume their data follows the same distribution.
- **Unbalanced:** Some clients may have a lot of data samples, whereas some may only own a single sample. This is also given for medical data, for example if a model is trained to combine data from hospitals and data from smartphones, where the number of patients in the hospital is very large, but each smartphone only collects data from a single person. Also just between hospitals, the number of patients for specific diseases can vary a lot.

Fig. 1 shows the training procedure of FL, which is reiterated later in Algorithm 1 (Section 3.2). Some institution or researcher (taking the role of the server) begins the process by initialising an ML model and sending all its parameters (denoted by θ^0 in Fig. 1) to each of the participating parties of the system. The goal of FL is then to find optimal values for the parameters, such that the ML model generalises well on the joint, federated database. In an iterative process, the server notifies a number of clients and provides them with the current model parameters (θ^{t-1}), which the clients use to overwrite their local model. Next, the selected clients partially train the model by using for example stochastic gradient descent (SGD), a common approach to converge to a minimum of the error on the local dataset (X_k, Y_k). After some predefined number of local training epochs, each client transmits updates to the parameters ($\delta\theta_k^t$) back to the server for aggregation. The final step is then to update the previous parameters (θ^{t-1}) by the average update received from the clients, which are weighted according to the number of data points for the individual clients (n_k). This process is repeated until the model has sufficiently converged and performs well for all clients.

There has been a lot of research into different FL training algorithms, communication protocols or attack and defence measures since 2016. Large companies like Google [34–37, 47, 48], Amazon [108] and Huawei [10] are driving the

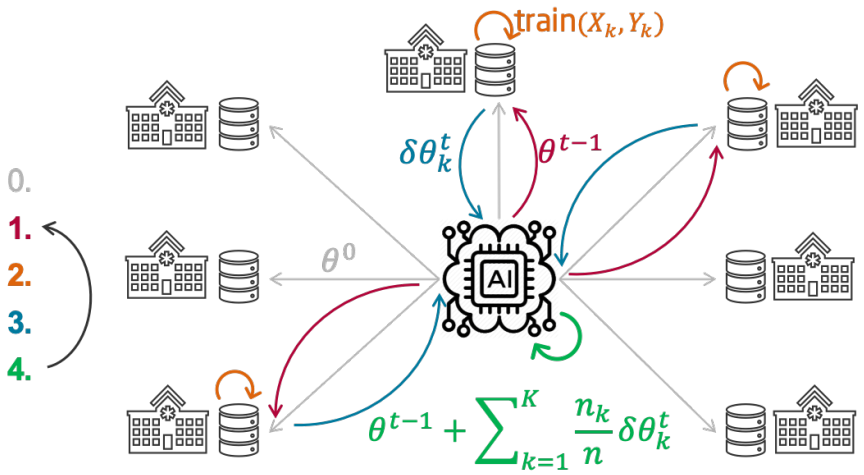


Fig. 1. Overall training process for federated learning. The initial model is distributed (0). Per global epoch, some clients are selected and receive the current parameter values (1). The selected clients update locally (2). The local updates are sent back to the server (3). The server aggregates all received local updates (4). Steps 1 through 4 are repeated until convergence.

research forward and are looking into this method for smartphone use or privacy-preserving user recommendations. In the last years, the amount of published papers in this field has increased drastically (see Fig. 3(b)).

There exist two previous narrative reviews about FL. In 2019, Yang et al. [74] gave a high-level introduction to the field of FL, the underlying privacy concepts, as well as related work and applications. They also identified the healthcare sector as a major benefactor. Secondly, Kairouz et al. [103] recently published an extensive review showing the advances, but also open questions for FL overall. We are aware of those works, and although both reviews are well-researched and provide valuable information, we found that the overlap of included papers is not that high: out of 80 papers considered in this paper, there are 13 also in the former, 23 in the latter review. Moreover, this paper follows a stricter, more systematic review approach using the PRISMA process [109] for paper selection and guidelines from BA and Charters [89]. We dive deeper into some of the proposed approaches and look more in-depth into the healthcare aspect of FL research.

The remainder of this paper is organised as follows: The research questions are introduced and the search process is explained in Section 2. Section 3 then shows the results of the search and Section 4 provides a discussion of the research questions. Finally, Section 5 concludes the paper.

2 METHODS

2.1 Research Questions

We aim to provide an extensive and structured overview of all papers relevant to FL which is stated in the first research question.

RQ1: What is the state of the art in the field of FL and what are its limitations?

Additionally, our goal is to show evidence that the medical field can benefit substantially by incorporating FL This motivates the second research question.

RQ2: Which areas of FL research are most promising for digital health applications?

2.2 Search Process

The literature search was performed over the time period from 01 January 2016 until 31 June 2019 using the ACM Full-Text Collection, arXiv, IEEE Xplore, PubMed and WebOfScience libraries. The general search terms used are:

"federated learning" OR *"federated deep learning"* OR
"federated machine learning" OR *"federated SGD"* OR
"federated optimi[sz]ation"

We did not include health-related terms in our search, because we are also interested in the general FL research, and health-related FL research is a subarea thereof. Since each library requires the search query to follow some specific rules, the exact query terms are listed in the supplementary material.

2.3 Inclusion and Exclusion Criteria

This review paper should provide readers with a good understanding of FL and a number of more in-depth descriptions about options on how to set up such a system. Moreover, the reoccurring theme is the use for the healthcare sector, and this we chose the following inclusion criteria. Included are papers which...

- consider FL at the centre of their research.
- use FL for training an ML model on medical data.

On the other hand, the surveyed query terms return many irrelevant works to this review, which lie out of scope or cover completely unrelated topics with simply mentioning FL a single time. Thus we excluded papers which...

- require participating clients to share their private data (encrypted or not).
- assume, clients possess IID data. This is not a realistic setting for real-world applications, especially for medical data.
- discuss federated reinforcement learning.
- don't present novel ideas, but simply describe an implementation of FL in some application (exception: medical application (RQ2)).
- describe a fully decentralised implementation of FL (e.g. Blockchain, Peer-2-Peer).
- cover a topic other than FL, i.e. unrelated papers mistakenly returned by the query.

Especially for the third and fifth entry of the above list there exists a lot of research. Although federated reinforcement learning it is an active sub-field of FL research, reinforcement learning and its applications are quite separate from un- and supervised learning, using very different underlying concepts. In addition, we found no paper looking into federated reinforcement learning for healthcare, thus we omitted this area of research.

A research area closer to FL is fully decentralised learning using a Blockchain or direct Peer-2-Peer network to exchange messages in terms of model weights. Although we found a paper discussing fully decentralised learning for medical data [119], we opted for excluding this area of research, because the traditional understanding for FL includes a client-server split and model aggregation on the server.

2.4 Data Collection and Analysis

In order to provide a numerical analysis of the reviewed literature and explain certain approaches in Section 3, we extracted information from each included paper and organised it in a spreadsheet.

The data extracted from each paper is:

- Title and year of publication
- Whether the paper
  - presents an FL training algorithm (Section 3.2, Section 3.1)
  - presents an FL security or privacy protocol (Section 3.4, Section 3.5)
  - presents an FL communication protocol (Section 3.3)
  - mentions health or medical use for FL (Section 3.6)
  - mentions differential privacy (Section 3.5.1)
  - mentions multi-party computation (Section 3.5.3)
  - mentions homomorphic encryption (Section 3.5.2)
- Empirically investigated
  - ML models (Fig. 3(c))
  - dataset(s)
- Research question / problem to solve
- Proposed hypothesis or solution
- Results and discussion

3 RESULTS

Fig. 2 shows the PRISMA flow diagram [109] which describes the process of searching, selecting and excluding papers. According to the inclusion and exclusion criteria from Section 2.3, papers were selected for full reading and out of the 167 initial papers, 80 were included in this review. The exclusion criteria and the number of papers excluded for each reason are listed in Table 1.

Table 1. Reasons for paper exclusion and number of corresponding papers

Exclusion Criterion	# Papers
Requires sharing private data	3
Requires IID data	4
Federated reinforcement learning	4
Only implementation of FL, or usage as tool	19
Fully decentralised method	9
Off topic	48

Numerical analysis of the included literature resulted in the following observations. First, Fig. 3(a) shows the search engines used to find the papers together with the number of papers included in this review. Note that the total number of papers is bigger than 80 since some included papers could be found on multiple search engines. More than half of the papers were found on arXiv, a platform without direct peer-review, which is not surprising since FL is a young and emerging research topic. Also, notably, only 4 papers on PubMed, a medical paper library, were related to FL, which points to a lack of papers about the usage of FL in the medical domain. On the other hand, there are 33 selected papers which at least briefly mention healthcare as a major beneficiary of FL application, encouraging more research into that direction.

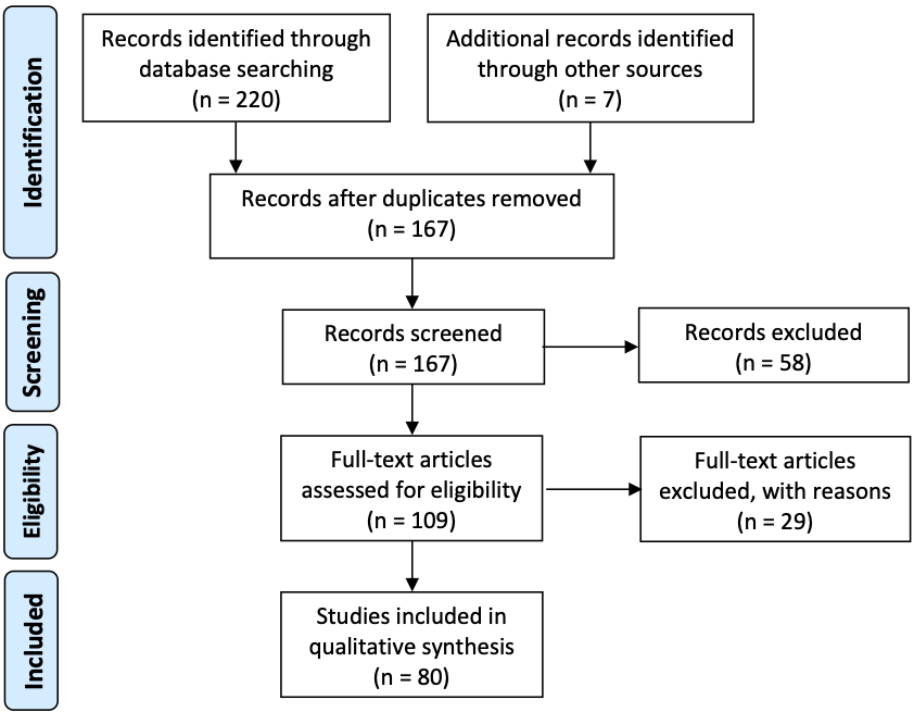


Fig. 2. PRISMA flow diagram

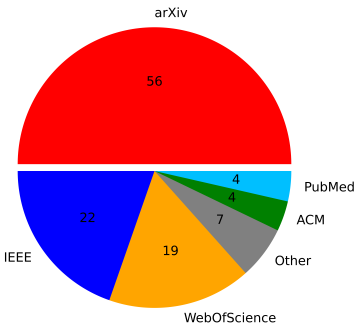
Fig. 3(b) illustrates the number of papers by publication year. Since 2016, the number of papers about FL has been steadily increasing and assuming the number of papers in the second half of 2019 increases linearly throughout this year, we can observe an almost exponential growth of the research field in terms of the number of published papers per year.

We also want to provide an overview of ML models that papers used for experimentation with, and evaluation of their FL approaches. The results are shown in Fig. 3(c), where convolutional neural networks (CNNs) are the most commonly used models, but also neural networks (NNs) and recurrent neural networks (RNNs) are quite frequent. Less explored are support vector machines (SVMs) and regression models. The *other* model types include tree-boosting systems and collaborative filter.

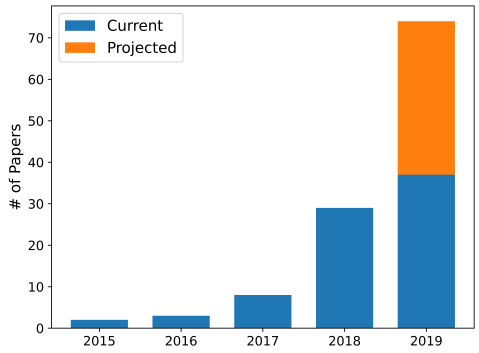
Fig. 3(d) shows the number of papers which deal with various defence concepts relevant to FL. All of those concepts will be properly explained in Section 3.5. We can already observe, that differential privacy is most commonly used.

Federated Averaging. In the introduction, we already explained the process of FL as proposed by McMahan et al. [47], which is widely considered as the initial FL paper (see Fig. 1). To recap, the goal is to train an ML model on a federated dataset. We will use as an example a neural network model, which is most common in papers included in this review. We will briefly introduce the training of neural networks in its most typical way.

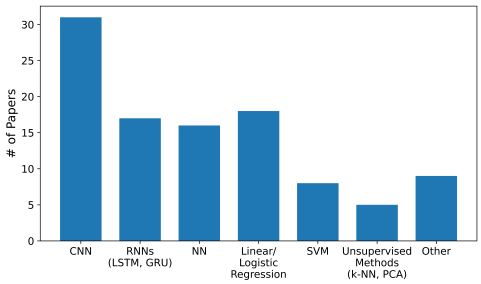
Neural networks make predictions by traversing a data point through a net of neurons, multiplying neuron inputs with the neuron weights and applying an *activation function* like a rectified linear unit (ReLU), until the last (*output*) layer determines the predicted label for the input. A well-trained model predicts the labels of all data in the training (and



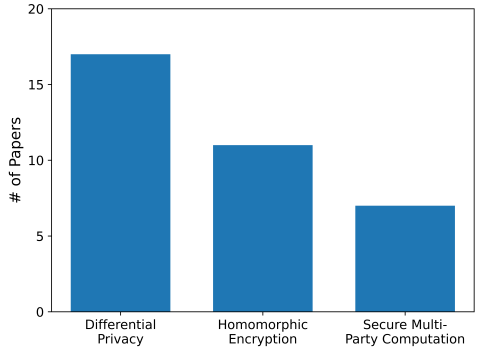
(a) Number of included papers, found on the most common computer science and medicine literature search engines.



(b) Number of included papers organised by the year of publication (as of 15 May 2019). The bar for 2019 includes an extrapolated number of papers that will probably be published during the rest of the year.



(c) Frequency of ML model types. Different kinds of neural networks (CNNs, RNNs, NNs) are the most common.



(d) Frequency of federated learning defence concepts (see Section 3.5)

Fig. 3. Quantitative analysis of reviewed literature

validation) set with sufficiently high accuracy, and in order to achieve that, a loss or error function (e.g. mean squared error (MSE) or cross-entropy error) is defined, which has to be minimised. Starting with random weight initialisations θ^0, the neural network receives a number of training samples (so-called *mini-batches*), calculates their prediction and the corresponding loss value. The gradient of the loss w.r.t. each weight ($\nabla L(b_k; \theta_k)$) is then propagated back through the network, and the neuron weights are updated according to SGD with learning rate η. One pass through the whole training dataset is called an *epoch* and after a few epochs, the weight values should have converged to a global (or local) optimum. The accuracy of a trained neural network can then be determined by presenting a test set of unseen data and measuring the fraction of correct predictions. [105]

The initial FL algorithm, as explained in the introduction, is shown in Algorithm 1. Although the formal algorithm in [47] suggests that the complete model weights are shared between clients and server, in reality, most approaches only share the parameter updates to reduce the amount of transmitted data and enable efficiency measures discussed in Section 3.3. In a more extensive empirical evaluation of the impact of the FL hyperparameters, McMahan et al. found

that choosing a fraction of $C = 0.1$ clients per round (out of the total K clients) is a good first choice when using local mini-batches, and smaller values are seldom good. Moreover, an increased number of local epochs E or similarly a reduced local batch size B can reduce the communication cost of the system and speed up global model convergence, given the clients have sufficiently strong computing machines. However, the effectiveness of this hyperparameter is reduced when the data is non-IID (see Section 3.1.1).

Algorithm 1 `FederatedAveraging (FedAvg)` [47]

Server executes:
  Initialise θ^0
  $m \leftarrow \max(C \times K, 1)$
  for $t = 1$ to T **do**
    $S^t \leftarrow$ (random set of m clients)
    for each client $k \in S^t$ **do**
      $\theta_k^t \leftarrow$ **ClientUpdate**(θ^{t-1})
    end for
    $\theta^t \leftarrow \sum_k \frac{n_k}{n} \theta_k^t$
  end for

ClientUpdate(θ): ▷ for client k
  $\theta_k \leftarrow \theta$
  for each local iteration E **do**
    for each batch b_k in client's split **do**
      $\theta_k \leftarrow \theta_k - \eta \nabla L(b_k; \theta_k)$
    end for
  end for
  return local model θ_k

The remainder of this section is split into six subsections: First, we discuss characteristics of federated datasets and approaches for working with specific types of data. Then, we present different learning algorithms in the context of FL. After that, we move on to papers dealing with communication efficiency, followed by attacks and defences relevant to FL. All of these sections include a subsection about implications for digital health to put the existing FL research into a medical context. Finally, we show literature concerning healthcare applications specifically.

3.1 Characteristics of Federated Datasets

Dealing with federated data from different sources comes with its unique challenges and stands in contrast to a centralised dataset that is simply distributed amongst worker nodes.

3.1.1 Non-IID data. First and foremost, one has to assume, that clients participating in an FL system may possess data following different local distributions. Still, the goal is to find a well-generalising model for all clients. Although the use of FL greatly improves the generalising performance of a model in the presence of non-IID data (compared to local models), its accuracy is still affected a lot by it. Using as an example the common MNIST dataset consisting of grayscale images (28x28 pixels) of handwritten digits, imagine a researcher only has access to images from class 0 and fits a classification ML model to it. If he would now encounter data from any other class, the model would perform very poorly (in fact it would only predict a 0-label). Only if a model is given data from all possible classes, it has the chance to make the correct predictions for all classes. Imagine now, there are more researchers, all having data from two of the

classes (classes on clients can overlap), if they join their efforts in an FL system, they will together be able to train a model that is far more capable for the MNIST classification task than any local model. This is a very common setting used in FL research to evaluate the algorithm's performance in the presence of non-IID data. We will from now on refer to this data distribution as *2-class non-IID* (according to [77]), and to the case of only one data class per client as *1-class non-IID*.

Zhao et al. [77] found that in the extreme case of 1-class non-IID the test accuracy of CNNs trained with `FedAvg` is affected by 11% in the case of the MNIST dataset, 51% for CIFAR-10 [104] and 55% for keyword spotting datasets. For 2-class non-IID, the negative effect is less but still ranges from 2% to 16%, depending on some training parameters. In real-world applications, data from different sources, like different hospitals, is likely to be non-IID, and there are some papers trying to improve FL specifically in these situations.

For instance, Zhao et al. [77] observed that if the data is IID, the learned model parameters in `FedAvg` are similar to those learned using centralised SGD, but they differ for non-IID data. The paper provides proof that one cause for this weight divergence could be varying initial weights on each client. Another is the earth mover's distance (EMD) [118] between the data distributions of each client and the global distribution. The proposed solution is sharing a fraction of data globally, reducing the EMD and in turn improving the achieved accuracy. Additionally, the server can pre-train the model on the globally shared data which jump-starts the learning process on the client-side. With those measures in place, the paper reports an improvement of $\approx$ 30% for CIFAR-10 in the 1-class non-IID case. However, this rarely is possible for medical use-cases.

Thirdly, Sahu et al. [55] (and similarly Yao et al. [75]) propose including an additional regularisation term during local training such that the solution space for weights is close to the global weights of the last epoch.

Eichner et al. [16] expect non-IID data in terms of temporal differences for newly collected data. If your algorithm uses data from all over the world, then somewhere it will be daytime, and somewhere it will be nighttime, which affects the amount or type of data encountered. That is why Eichner et al. propose splitting training into multiple blocks and using a consensus algorithm to find an ideal model. The regional characteristics may influence the data as well, which is the core of Hu et al. [28]. Their algorithm clusters geographically close sites together and trains sub-models which can then inform the others for more generalised models for each cluster. This can be an important step towards working with global, non-IID medical data.

A way to deal with unbalanced datasets was proposed by Duan [15] and relies on data augmentation. Clients are categorised as *uniform* clients, if they possess enough balanced data, *slight* ones, if they only have a small local dataset, and *biased* clients, if their dataset is imbalanced. Mediators between client and server are responsible for the different groups and ask clients to perform data augmentations including random shifts, rotations, shears and zooms. This way, the global dataset is balanced, the training process is more stable and the resulting model performs better.

3.1.2 Vertically Split Data. When analysing papers on FL, one has to not only look at situations in which the data is horizontally split but also those in which data is vertically split (see Fig. 4) [74]. A horizontal split is given if all clients own data in the same feature space, but have (mostly) different samples. As an example consider multiple hospitals from different countries as clients, who will collect very similar data but have little to no overlap of patients. On the other hand, data is vertically split amongst clients if they own different features but from the same sample space. For instance, imagine a hospital who might refer many patients to a specific cardiologist. Both hospital and cardiologist collect different kinds of data but will have many patients in common. If both splits are partially given, so there are some of the same samples with some of the same features, but also samples and features that don't overlap, this is a

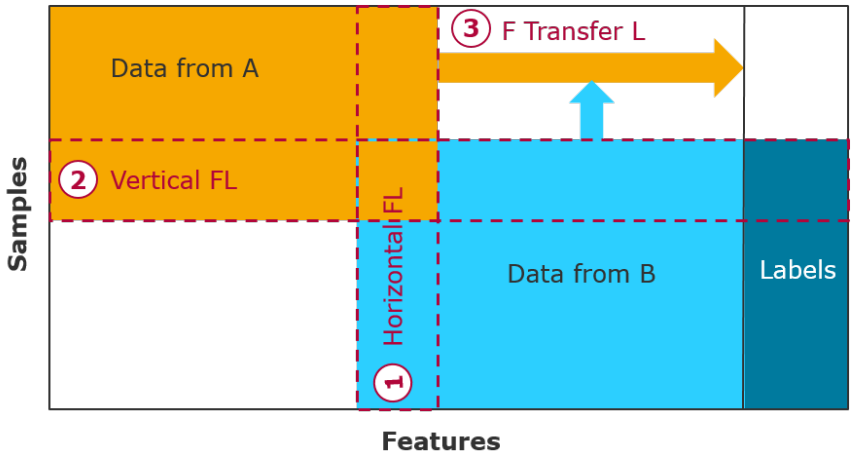


Fig. 4. Types of federated learning (based on [74]). Data from clients can overlap in the feature space while varying in the sample space (1), it can contain the same samples, but varying features (2), or there may be some overlap as well as separation in either dimension (3)

situation where federated transfer learning can be applied [45]. Most research is concerned with horizontally split data and we will separately cover the papers on vertically split data in Section 3.1.2.

As illustrated in Fig. 4, data is considered to be vertically split if multiple clients have data from the same samples, but different features. Often, only one of the clients has access to the label of a data record, making it even harder to jointly train a model. The main issue for vertical FL is finding a privacy-preserving way of matching equal samples across clients, which is termed *entity resolution* or *record linkage*.

In [42], Li et al. developed an algorithm for *VERTIcal Grid lOgistic regression*, applicable if all clients have access to the label for a particular data sample. They formulated a dual optimisation problem for logistic regression, that is solvable by constructing the global gram matrix, which preserves data privacy, and optimising with Newton's method. The accuracy of the model was reported to be comparable to a centralised alternative, however, the approach is limited by the number of samples m, since it includes inverting a large Hessian matrix, which has complexity $O(m^3)$, and VERTIGO may not converge given highly imbalanced data.

Finally, Cheng et al. [13] describe a tree-boosting system called *SecureBoost*, where the party holding the label for each federated sample is called *active party* and takes the place of the server, while all other parties are *passive parties*. Like Hardy et al. [24], the approach uses homomorphic encryption to resolve entities across different parties and training of the ML model is lossless in the sense that the training loss is the same as the loss of a model trained on a centralised dataset. Specific for this tree-boosting model, the authors claim that their approach is private given that the active party does not collude with a passive party, however, they require revealing an entity ID.

Overall FL in the presence of vertically split data is by far not as explored as for horizontally split data, and it is limited by the number of samples, features and collaborating parties. The entity resolution relies heavily on encryption to preserve privacy, which requires additional computational power, which might not always be given.

3.1.3 Implications for Digital Health. In healthcare, both non-IID and vertically split data is quite common. This follows from the fact that over two thirds of the FL papers concerning digital health specifically assume either non-IID or vertically split data [7, 12, 30, 31, 39, 42, 60, 61]. We already motivated the medical use-case for a vertical data split above,

however, we would like to point out, that even though lots of medical data is distributed vertically, a small amount of contributing hospitals and clinics would probably result in only a few entity resolution matches. A potentially interesting scenario to investigate could be a vertical combination of health-related data from patients' personal devices and hospital data. Different data distributions amongst hospitals can for example occur when hospitals have a specialisation, such that some medical procedures are more common than others. Alternatively, hospitals around the world also encounter different population structures, meaning ML have to address non-IID data.

3.2 Learning Algorithms

This subsection gives an overview of research towards optimising the learning algorithm by McMahan et al. or proposing a new one.

A lot of the groundwork for [47] is explained by Konecný et al. [34] and later in [35], co-written by some of the same authors. They explain the issues for working with many distributed datasets in the wild (see Introduction (Section 1)). The paper investigates situations where client data is sparse (i.e. only a few features are present per client), because prior distributed ML solutions are not sufficiently accurate. With stochastic variance reduced gradient federated optimisation (`SVRGfo`) the update rule for local parameters is changed, such that it keeps in mind the loss of the previous model for the whole dataset, which is computed collectively in each global iteration. Additionally, the paper introduces two new parameters which indicate sparsity of the data by modelling appearance and frequency of features per client.

Multiple papers claim that averaging client parameters weighted simply by the amount of training data available for each client (Algorithm 1, Line 9) is not good enough [11, 32, 40, 50]. For example, the global aggregation step can keep previous updates in mind, as suggested by Leroy et al. [40], by using the exponentially-decayed first- and second-order moments to update per parameter.

A training algorithm called `SplitNN` which deviates a bit more from the standard FL is proposed by Vepakomma et al. [65], who suggest splitting a neural network such that some layers are trained by clients and some are trained by the server. A similar path is taken by Wang et al. [66]. The motivation for this approach is limiting the computational effort for clients, which might not have the most recent computers, like smaller hospitals. During training, clients calculate the output of the last local NN layer (the *cut layer*) and send it to the server, which in turn completes the forward pass through the network, calculates the loss and propagates it back to the clients. This setup requires the labels to reside on the server and thus some kind of entity resolution (see Section 3.1.2). As an extension, the paper proposes a U-shaped structure of the NN, such that the middle layer(s) are trained on the server, but the output of the server cut layer is transmitted back to the clients. This provides a stronger privacy guarantee and is more applicable in medical applications. In experiments with a CNN to predict images from the CIFAR-10 dataset, the required computation was significantly lower than for `FedAvg` and the used communication bandwidth was also lower compared to `FedAvg`, particularly when the number of clients was large.

3.2.1 Hyperparameter Optimisation. A critical part of training more complex ML models like CNNs is the tuning of hyperparameters like the learning rate for SGD, the network structure in terms of the number of layers and neurons, or any other parameter included in the loss function. This is difficult for an FL system since there is no possibility to try out different hyperparameter combinations on a given dataset. If there exists an open-access dataset that is similar to the data encountered by clients, it is possible to use this for some initial experimentation, but the values will not be as good, as if the hyperparameter optimisation could be done on the actual data. Thus there are some papers looking at this issue and possible solutions.

First, Zhu and Jin [79] introduce a genetic algorithm which starts with a *population* (basically a set) of possible hyperparameters and *evolves*, i.e. optimises them during federated training to reach to a combination of hyperparameters that work well for the given data. They optimise both for minimal model loss, as well as a low model complexity to find a well-generalising model setup. The limiting factor for this approach is the amount of training required per client since they will have to optimise multiple models simultaneously. If sufficient computing resources are available, the genetic algorithm is able to find a model with high accuracy and a lower amount of model parameters than the compared model baseline reported in other papers.

A different way of handling hyperparameter optimisation is proposed in [61]: the Restricted Federated Model Selection (`RFMS`) algorithm by Sun et al. [61] restricts training of the model to a single site while using all other sites for adapting the hyperparameters. In a Bayesian optimisation pattern, different clients can propose a new set of hyperparameters, informed by previous trials and reported accuracies from all clients. The following two paragraphs consider the optimisation of a single hyperparameter.

Learning Rate. In an attempt to avoid local optima in the global model, Xu et al. [71] employ a cyclical learning rate per client that is reset in each round of communication, as well as an increasing number of local training epochs over time, when the magnitude of local updates is very small. Koskela and Honkela [38] have a different reason for changing the learning rate during training. They argue that differential privacy (see Section 3.5.1) prohibits training a model for a large number of global epochs, which makes converging as fast as possible a high priority goal. They start the FL with an initial guess for the learning rate and try both taking one step with that learning rate, or two steps with half the learning rate. If the loss difference exceeds the tolerance hyperparameter, the learning rate is updated.

Epochs. Finally, Huang et al. [31] look at the number of local epochs to optimise during training. Their Loss-based Adaptive Boosting Federated Averaging (`LoAdaBoost FedAvg`) algorithm is motivated by the heterogeneity of local datasets (see Section 3.1.1) and aims at boosting the training process of slower learners by increasing the number of local epochs for them. Every client has to report their local loss to the server, which then sends the median loss to each client. Whenever a client does not improve on that median loss after the local training procedure, they train for a few more epochs and check again. This is capped at double the initial local epochs.

3.2.2 Asynchronous Training. If there are clients in the FL network which have a bad internet connection or very limited computational capacity, these so-called *stragglers* might hold up the training progress. Thus some papers propose an asynchronous training algorithm in which the server allows feedback from clients at any time and schedules its requests accordingly.

One such idea can be found in [70], describing an algorithm called `FedAsnc` to schedule optimisation in an asynchronous fashion, weighting each client's contribution to the global objective by a measure of staleness. The server runs two threads in parallel: the scheduler thread is responsible for triggering the training process on selected clients, while the updater thread accepts parameters from clients and updates the global model. They show that the approach improves efficiency, flexibility and scalability compared to the initial `FedAvg` and even in the worst situation, with a high overall staleness in the network, their convergence rates are similar.

One further step of asynchronism is described by Chen et al. [11], where not only the global model is updated in an asynchronous way, but also deep and shallow layers of the model. They base their *temporally weighted asynchronous federated learning* (`TWAFL`) algorithm on the observation about deep neural networks that shallow layers tend to learn general features which might be applicable to multiple similar datasets, whereas deep layers learn much more specific

features related to the current dataset. Thus their proposal is to train shallow layers in every iteration, but skip a number of rounds before updating the deep layers as well to keep a good ability to generalise.

3.2.3 Fairness. One distinctive feature of training in an FL system is that the resulting model will not perform as well for some clients as for others which is due to varying data distributions. If one client's data is widely different to everyone else's, his updates will likely be overruled by the majority of different updates received from other clients. Then the resulting model is of little use to him. In that case, one should simply exclude the outlier from FL and let him train a local model, but generally there is still variation in the model accuracy for different clients.

Mohri et al. [50] build on this idea by proposing an agnostic federated learning (AFL) algorithm which models the clients as a mixture and introduces additional mixture weights per client into the overall loss function. This makes the learning algorithm favour the client with the highest loss, while slightly degrading the model performance for the most benefitting clients. The paper includes theoretical proof for their approach and the algorithm can also be used for domains similar to FL, such as cloud computing.

While the previous paper chooses to use the worst client as the benchmark for fairness, Li et al. [41] optimise the variance in accuracy for clients. Their q-Fair federated learning (q-FFL) approach introduces the fairness hyperparameter $q \in [0, 1]$ into the global objective function, where higher values correspond to more fairness more equivalent to AFL, and setting q to zero is just FedAvg. Li et al. give hints for choosing a suitable fairness value, and in comparison to AFL, q-FFL achieves better accuracies overall in a lower number of local epochs.

3.2.4 Resource Constraints. Some papers do not only consider the ideal training procedure for clients with the same resources but instead model their constraints in terms of battery power, computation speed or internet connection.

Wang et al. [68] opt for setting a time budget for local optimisation and global aggregation. They define a control algorithm which solves the learning task given those constraints by adapting the total number of training epochs and the frequency of aggregation steps. The paper includes theoretical proofs for their approach, as well as evaluation using four different datasets and models.

Putting more focus on the clients' resources, Nishio and Yonetani [52] propose the server asks clients about their capacities, before making an informed selection which clients to choose for a particular global epoch. The goal is to include as many clients as possible in each round of communication while not exceeding the time threshold.

Xu et al. [72] provide a more detailed model, where a client's utility is calculated using information about local dataset size, battery charge, CPU cores and frequency. This algorithm relies on payment of the server for the usage of clients' resources. This means, that clients bid with their resources on the chance to take part in the training and receive a reward.

Finally, Zou et al. [80] use an evolutionary game model to find the optimal trade-off between model accuracy and energy consumption of clients. The algorithm they propose is proven to find an equilibrium for the given model, such that the benefit and cost of training are chosen optimally.

3.2.5 Federated Multi-Task-/Transfer-/Meta-Learning. Multi-task learning is used to train multiple related tasks simultaneously, while their relationship is modelled with e.g. a matrix. Smith et al. [59] first proposed a federated version of multi-task learning, where the algorithm alternates between optimising the multi-task goal and the relationship matrix. Their MOCHA algorithm can train convex models like SVMs and the paper includes several considerations about communication cost and fault tolerance in the federated setting. For non-convex models like NNs, Corinzia and Buhmann [14] developed a VarIational fedeRaTed mUlti-tAsk Learning (VIRTUAL) algorithm. The multi-task alignment

is performed with lateral connections between the client and server NNs, and the server's parameters are updates using approximated variational inference.

Related to multi-task learning is transfer learning. Here, existing models from a related domain are retrained to reach an appropriate model for a given domain. This way, a high performance model can be found for different, but related datasets. Liu et al. [45] propose a federated transfer learning algorithm with provable security guarantees and better model accuracies than simple locally trained models. Similarly, Chen et al. [12] adopt a transfer learning approach to personalise a human activity recognition (HAR) model to individual subjects.

A meta-learning approach is described by Chen et al. [10] which does not learn the ML model directly, but instead leans a meta-model that can be quickly trained to the model required by each client. The advantage is that the meta-model can be kept a lot smaller than the actual model, and thus the transmitted and stored data for the model is reduced. The area of application in the paper is a recommendation meta-model which could be trained efficiently and accurately.

3.2.6 Federated Filtering and Matrix Factorisation. While most papers on FL discuss algorithms and solutions for (deep or recurrent) neural networks, Ammad-ud-din et al. [2] developed a federated implementation of a collaborative filter [93]. This model relies on computing a user-item/rating matrix, which can be of very high dimensionality, which is why it is split up in the multiplication of two lower-dimensional matrices yielding a latent representation in between. Their experimentation section shows that for a synthetic, the Movie-Lens and In-House datasets the filter performs very similarly to one trained on centralised data measured in terms of five different common ML metrics.

Three other papers incorporate a filtering or matrix factorisation model. Chai et al. [9] He et al. [26] focus on the security and privacy aspect in their algorithms, whereas Sanyal et al. [56] investigate specifically the application of a federated filter for Internet of Health Things (IoHT) edge devices collecting data over time.

3.2.7 Implications for Digital Health. Almost all papers on FL for digital health can be categorised as FL algorithm research. The reason could be the novelty of the research area where at first the basics have to be comprehended before research can fan out to communication efficiency (Section 3.3) or adversarial FL (Sections 3.4 and 3.5).

FL with hospital data could negate the issue of hyperparameter optimisation since hospitals usually have a large database already that can internally be used to tune the ML model before sharing it with other medical participants. The strategies explained above to adapt local learning rates and the number of epochs can improve model performance, which is key for medical use-cases.

Additionally, fairness is incredibly important in medicine. There have been several discussions about biases against ethnicities or subgroups in medical ML models which have to be avoided [98, 112]. Consequently, the model performance should not vary too much between different learning participants who might have different patient data distributions.

Many medical use-cases for FL may be related, like for instance medical image segmentation, object detection and classification or patient mortality and discharge time. This motivates the use of transfer- or multi-task-learning to save time and computational power for solving related problems which, on the other hand, is lacking research so far.

3.3 Communication Efficiency

FL requires clients to repeatedly send their model parameter updates and in return receive the new global parameters. Especially when there is a large number of clients (>100) involved in training, the communication efficiency becomes the main bottleneck of FL to achieve a quick model convergence [8, 22, 43, 50, 52, 59, 62, 75, 78][103]. This motivates a branch of FL research looking into ways to improve the efficiency of information exchange between the participants.

Some aforementioned learning algorithms do already improve the communication efficiency implicitly by improving the training progress made per round of communication [31, 38, 55, 71, 75].

As a good starting point for the now following papers, Konecný and Richtárik [37] show different encoding, communication and decoding protocols for the purpose of estimating the mean of several distributed values, as it is required for the averaging step in FL. The paper formally derives bounds for the MSE of the various mentioned protocols and provides formulae to compute the optimal MSE given a communication budget.

3.3.1 Gradient Compression. Approaches that aim at reducing the amount of transmitted data per weight update up-/download are called *gradient compression* methods. Konecný et al. [36] proposes distinguishing between *structured* and *sketched* updates, where the former restrict the parameter space and the latter compress the parameters to allow for more efficient encoding and communication. The methods are not mutually exclusive and can be combined if the goal is to optimise communication efficiency at the expense of model accuracy.

Structured Updates. This type of gradient compression restricts the parameter space so that they can be encoded with fewer bits than full parameter updates. Konecný et al. [36] propose constraining the matrices to be of *low rank*, which reduces the communication cost inversely proportional to the defined maximum rank, or to follow a sparse matrix *random mask*, such that the number of transmitted parameters is smaller.

Another idea by Caldas et al. [8] is *federated dropout* where clients train sub-models of the global model, defined by zeroing out a fixed number of fully-connected layer activations or convolutional layer filters selected at random. The server can then reassemble the complete model and average the transmitted gradients from multiple clients.

Sketched Updates. In contrast to structured updates, sketched updates first compute the complete gradients for the model parameters, but then compress them to be more efficiently encoded and transmitted. One possibility is *subsampling*, selecting a random subset of updates (per client) to communicate, which after averaging still gives an unbiased estimate of the true average update [36]. Alternatively, *probabilistic quantisation* compresses every scalar x of a vector $\vec{v}$ (or of column vectors of weight matrices) to either the maximum or the minimum coordinate value with probability $\frac{x-\vec{v}_{min}}{\vec{v}_{max}-\vec{v}_{min}}$ or $\frac{\vec{v}_{max}-x}{\vec{v}_{max}-\vec{v}_{min}}$ respectively. The effect of quantisation can further be improved when multiplying vectors with a rotation matrix before quantising, and performing the inverse rotation before aggregation on the server-side [36]. Similar results were found by Suresh et al. [62] and proven in their mostly theoretical paper.

Probabilistic quantisation is also used in the *lossy compression* approach by Caldas et al. [8], but here, first, the weight matrices are reshaped into vectors and a basis transform is applied which serves a similar purpose as the rotation in the previous paper, optimising the amount of information retained after quantisation. Next, a fraction $1 - s$ of coordinates are set to 0 by subsampling uniformly at random, before the same probabilistic quantisation is applied as in [36]. In contrast, this compression mechanism is used for server-to-client communication and not the other way around. The authors report a reduction of up to 14× without degrading model accuracy.

Lin et al. [43] developed a composite compression mechanism, called *Deep Gradient Compression (DGC)*, consisting of four methods: *gradient sparsification*, *momentum correction*, *local gradient clipping* and *warm-up training* (explanations can be found in the paper). Empirically, Lin et al. found a compression ratio between 270 and 600 without impacting the accuracy.

Gradient Upload Filter. A final option to reduce network traffic, which does not directly compress gradients, but instead excludes some clients from uploading their updates has been proposed by Wang et al. [67]. They suggest comparing the newly computed local weight updates with the global update from the previous epoch. The metric

for choosing whether to upload an individual client's gradients to the server is simply the percentage of same-sign parameters in the two updates, determining how *aligned* the one update is with the other. If this percentage lies below some predefined threshold, the client will discard his update, which, claimed by the authors, has the capability of reducing the network footprint by a factor of up to 14.

3.3.2 Wireless Channel. A niche of FL research looks into ways of using the wireless channel between clients and server more efficiently. This is especially important if clients have limited resources and potentially a bad wireless connection, but the model training has to be as fast as possible. This research sub-area is very technical and less concerned with improving FL, but more with network channel specifics. Moreover, it is not so relevant for medical applications of FL, since they rarely rely on the benefits of optimised wireless networks. Thus we only mention the papers Amiri and Gündüz [1], Feng et al. [17], Tran et al. [63], Yang et al. [73] and leave further investigation of the methods to the interested reader.

3.3.3 Implications for Digital Health. So far, the medical field is lacking experimentation about the trade-off between compression to improve communication and model accuracy. The fact that ML models in digital health are supporting mechanisms for important decisions which could impact patients' lives means that model performance is the most important factor. In addition, the existence of large databases for initial model training, could make communication improving strategies for healthcare superfluous, because there is no need to incorporate new data into the model as quickly as possible. However, more real-world evaluation is needed.

3.4 Attacks

A large part of FL research is concerned with the security and privacy of the algorithms. Before diving deeper into the specific approaches, the following subsection will outline a taxonomy for the kinds of adversaries relevant for FL, and their capabilities. Afterwards, we present four different attack types and more in-depth algorithms in literature.

3.4.1 Taxonomy. The attacks can be categorised in three axes [51] [92]:

Attacker Role: First of all one has to distinguish between an *adversarial server* and one or multiple *adversarial clients*. A malicious server has a lot more capabilities than a client, like isolating clients and attacking each individually, however an adversarial client could potentially control multiple other client devices as well, which is called a *Sybil attack* [95][19].

Attacker Capabilities: When discussing the potential of an attacker, there are mainly two variants. Either the attacker is *honest-but-curious* (also *semi-honest* or *passive*), where he follows the definition of the training protocol but tries to gain as much information as possible by analysing all data he receives, or the adversary is *malicious* (or *active*), meaning he applies whatever means necessary to attack the system or its participants.

Attacker Knowledge: For this axis of the taxonomy, the two extreme cases (and most common ones) are *white-box* or *black-box* knowledge, which can be applied to either data or ML model. In the former case, the adversary has complete knowledge over e.g. the model and in the latter, it is completely hidden to the attacker. In between, there are a number of *grey-box* scenarios that can be defined on some sub-knowledge of the attacker.

3.4.2 Membership Inference Attack (Tracing). In these types of attacks, the adversary's goal is to predict whether a particular (known) data sample was used during model training. Existing algorithms so far rely on training an attack model in the form of a binary classifier that, given a data sample and parameters from the attacked model, predicts whether or not the sample was used for training. One limiting factor of membership inference attacks is the requirement

of an auxiliary dataset from the same (or a similar) distribution as the training data, which is labelled for the adversarial objective, in order to train the attack model. Nasr et al. [51] report an attack accuracy between 62% and 86% for a passive adversarial client or server, which can be increase to up to 93%, if the adversary crafts his parameter updates using gradient ascent (as opposed to gradient descent) on the investigated data samples. In a slightly different kind of membership inference attack, Melis et al. [49] try to infer features or properties of train samples, instead of the membership of a specific sample.

3.4.3 Reconstruction Attack. This attack is aimed at breaching the training participants' privacy by reconstructing their data samples (or very similar ones). It has been shown that the gradients transmitted by clients leak information about the underlying training data. Like for membership inference attacks, the adversary trains an attack model on the side, but in contrast, for reconstruction, this is some kind of generative model.

The most simple reconstruction attack is a *model inversion* attack [97], which describes the process of reverse-engineering data samples by observing gradients in the model after training on samples. However, this approach only gives representatives for whole batches and thus may be quite noisy, particularly for client attackers in an FL system. So there is some research about more advanced attacks of this kind.

For instance, Wang et al. [69] developed an attack based on generative adversarial nets (GANs) [99], which allows an adversarial server to reconstruct samples from an individual target client. The approach, termed *mGAN-AI*, builds a multi-task GAN in which the discriminator solves three tasks: discriminating real from fake samples, categorising the data classes, and identifying the target client. That means in turn, that the generator requires not only noise but also a sample class and client identity as inputs. The paper proposes using the same model structure for the discriminator as for the model trying to be trained by FL (except for the output layer). Since the server does not have access to any client data to train the GAN, *data representatives* are calculated by minimising the distance between parameter updates received from clients and parameter updates calculated using the data representatives, regularised with a measure of neighbourhood distance. In an active attack, where the server isolates the target client, the algorithm is simplified and the attack improved.

An active malicious client can also launch a reconstruction attack on a specific class in a similar way as explained above. Hitaj et al. [27] also propose using a GAN where the discriminator component mimics the FL model. After training the generator component to create samples of the attacked class with sufficient similarity, the adversary adds them to his local dataset, labelled as a new class. Consequently, in the following epochs, other clients have to work harder to make the model distinguish between those two classes, which reveals more information about the attacked class and improves the discriminator, and in turn also the adversary's generator.

3.4.4 Model Poisoning Attack. In contrast to the two previous attack types, a model poisoning attack does not attack data privacy, but instead the model itself. By introducing malicious samples in the training set of a model, the adversary can follow one of two objectives: either he is trying to achieve misclassifications for a single or small set of samples (*targeted attack*), or he wants to reduce the model performance overall (*indiscriminate attack*). In the context of FL, the attacker is assumed to be one of the clients, since the model owner is interested in good model performance.

A common way to poison a model in a targeted way are *label-flipping attacks* [91] and *backdoor attacks* [3]. The former works by flipping the label of samples of one class to a different one. When such samples are included in the training set of a classifier, it results in misclassification of the one class. While this targets a broad region of the data space, backdoor attacks aim at introducing a very specific misclassification of a single or a few samples depending on particular features or patterns. A possibility to find the required adversarial updates is using existing data poisoning

methods, such as *Fast Gradient Sign Method* [100] or *Deepfool* [111], and training the local model with them to get to the corresponding model parameters.

Bhagoji et al. [4] investigated the effectiveness of different poisoning attack strategies, from simple boosting of the malicious updates to increase their impact, to alternating minimisation of benign and adversarial training objective in order to improve resilience against outlier detection defences. Finally, they also looked at estimating the updates of other participants of the system, to include this additional information in the crafting of malicious updates.

The attacker can try to increase his influence by using *sybils*, which are additional participants of the system controlled by the same adversary. Looking at a label-flipping attack, controlling 2 out of 12 clients in an FL system can achieve close to 100% misclassification for the targeted label [19]. An option for mitigating this kind of Sybil attack was developed by Fung et al. [19] and is called `FoolsGold`. It works by calculating the cosine similarity between updates from different clients. In the update aggregation step of FL, similar updates are scaled down to counteract the influence of sybils. The authors found empirically, that their approach does not affect the accuracy of the model if it is not under attack, but can limit the impact of poisoning with sybils.

In addition to `FoolsGold`, there is a large body of research for defences against poisoning of ML systems. Known defences such as `RONI` (Reject On Negative Impact) [90] or `TRIM` [102] can be deployed by the server instance to protect the model and have been considered by papers of the FL domain [18, 19].

3.4.5 Linkability Attack. One more, recently considered goal for an adversarial server is linking updates from multiple clients in an FL system to the same person. For instance, one person could own both a smartphone and a tablet and use both devices to contribute to the same FL model. Then the server could be interested in linking updates from both devices to increase his knowledge about that one user and improve follow-up membership inference or reconstruction attacks.

The only paper in this review considering linkability attacks is by Orekondy et al. [54]. They rely on the idea that model updates show certain patterns characteristic for the person. The paper makes a further distinction between *identification attacks* and *matching attacks*. The former aims at, given a set of weight updates and their corresponding client IDs, identifying the client for a newly observed weight update using an ML model. This does not rely on information such as IP address, which could be avoided by routing network traffic through an anonymity network like *Tor*, as suggested by Hartmann et al. [25]. The matching attack, on the other hand, is then used for linking two updates to the same person. Again, the tool for this task is an attack network. Orekondy et al. report an area under the precision-recall curve (AUPRC) of well over 90% for the matching attack. Some possibilities for mitigating a linkability attack are differential privacy (see Section 3.5.1), replacing parts of the local data with publicly available datasets, or augmenting the data.

3.4.6 Implications for Digital Health. Vulnerability analysis is critical before deploying FL for real-world applications such as healthcare. Out of the reviewed papers about attacks on FL, none used a medical dataset directly, but medicine is mentioned as a potential field of application.

On the one hand, membership inference and reconstruction attacks could be detrimental if private patient data leaks to adversaries. On the other hand, model poisoning attacks, when undetected, can result in incorrect ML model outputs leading to illnesses not being detected correctly, unsuitable treatments being proposed or other potentially devastating consequences.

There are two scenarios considered in digital health research: either the FL system is to be deployed between medical professionals like hospitals and clinics, or the research includes data collected by patients directly, like with their

smartphones or wearables. In the former case, we can assume that all participants have good intentions and will neither try to attack the model (since they want to benefit from it) nor try to invade patient privacy. In addition, hospitals will have a large dataset, meaning that transmitted model gradients include aggregated information from lots of patients, such that individual privacy violation is more difficult. The latter case, however, requires more careful consideration of defence measures, because not every patient can be trusted. It is also easier to find out private information about individuals because they improve the model with their data alone with no previous aggregation. Nevertheless, this use-case for FL in digital health should not be overlooked, because there is a lot of potential in using data collected by patients at home and on the go, which is usually hard for doctors to gather.

3.5 Defences

Since a central goal of FL is to improve data privacy, the aforementioned attacks threaten to remove that benefit. Therefore, a number of known strategies have been adopted into FL algorithms to improve the resistance of the systems against them.

3.5.1 Differential Privacy. Differential privacy originated from data science and is used to describe how resilient a database and analysis thereof is against membership inference attacks.

Definition 1 ([87]). A randomised mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \subseteq \mathcal{R}$ it holds that

$$\Pr[\mathcal{M}(d) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(d') \in \mathcal{S}] + \delta$$

Adjacent inputs in the context of ML are defined as two datasets X, X' differing in a single training sample, such that $X = X' \setminus \{x_n\}$. The idea of the randomised mechanism is that the output of the ML model cannot be traced back to the impact of a single data point.

One way to guarantee differential privacy is to use a *Gaussian mechanism*. Assume we want to secure some deterministic real-valued function $f : \mathcal{D} \to \mathbb{R}$. Then we can use the Gaussian mechanism as follows:

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}(0, S_f^2 \sigma^2)$$

Here, S_f is the sensitivity of f, which is defined as the maximum absolute distance $|f(d) - f(d')|$ of adjacent inputs d and d'. When using this added Gaussian noise once together with f, it satisfies (ϵ, δ)-differential privacy if $\delta \geq \frac{4}{5} \exp(-(\sigma\epsilon)^2/2)$ and $\epsilon < 1$. For a more detailed description of differential privacy and differentially private SGD we refer to Abadi et al. [87].

In order to keep track of the privacy spendings over time, Abadi et al. [87] propose the use of a *moments accountant*, adapted from the privacy accountant in [107]. It is a procedure with the purpose of accumulating the privacy loss and its moments during training and evaluation of the ML model. If the privacy moments threshold is exceeded under (ϵ, δ)-differential privacy, the moments accountant notifies the participants and stops the training process.

There are a number of papers looking specifically into differential privacy for FL, the first of which is [21]. In order to scale the Gaussian mechanism to the dataset, the authors suggest scaling parameter updates using the formula $\Delta\overline{\theta_k} = \Delta\theta_k / \max(1, \frac{||\Delta\theta_k||_2}{S})$. Then the sensitivity is bounded by S and thus the Gaussian mechanism can use the noise distribution $\mathcal{N}(0, S^2\sigma^2)$. The paper also includes hints for choosing the right parameters for the given setting. One result of the experimentation section is that the cost of using differential privacy is negligible if the number of clients is

large enough (e.g. $K = 10000$), however for a relatively small set of clients, the privacy budget is spent quickly and thus impacts the achievable model accuracy.

McMahan et al. [48] also use a similar clipping method, but they propose some additional steps to best use differential privacy together with FL. First, they randomise the fraction of clients selected per round, then, after update clipping, the weighted averaging is exchanged with a different average estimator, and finally the Gaussian noise is added to the parameters.

Differential privacy can also help against reconstruction attacks, as shown by Bhowmick et al. [5] who split weight update vectors θ_k into their direction $\theta_k/||\theta_k||_2$ and their magnitude $||\theta_k||_2$. For this case, the authors introduce the notion of *(ϵ_1, ϵ_2)-separated differential privacy*, which together gives $(\epsilon_1 + \epsilon_2)$-differential privacy. The paper describes randomly choosing each client with probability C, as described in the previous paragraph. Afterwards, clients add noise to both the direction and magnitude of the parameter updates and transmit their product to the server.

3.5.2 (Additively) Homomorphic Encryption. Encryption can be used to trade-off computation time and privacy/security. Typically a party can follow a key-*generation algorithm* to find a private and public key for encryption. Then anyone can use his public key and the *encoding algorithm* to compute an encrypted representation of a value he wants to send to that party. The actual shared value can only be reconstructed using the *decoding algorithm* that requires the party's corresponding private key, which is why it has to be protected by all means.

When using encryption in multi-party systems it may be a desirable property of the encryption scheme to allow valid computations with the cyphertext, so that parts of the computation can be outsourced to other parties while preserving data privacy. Those types of encryption schemes are termed *homomorphic*, and they are further split in *fully* and *partially* homomorphic encryption schemes [116]. Fully homomorphic encryption preserves the validity of any kind of computation with encrypted values, whereas the partial relatives limit the possible operations. Additively homomorphic encryption is partially homomorphic and allows the addition of cyphertext as well as multiplication of an encrypted value and a clear value. Formally, additively homomorphic encryption allows the following (where encryption of a value is denoted by $[[\cdot]]$):

$$[[x]] + [[y]] = [[x + y]]$$

$$[[x]] * y = \sum_{i=1}^{y} [[x]] = [[x * y]]$$

Subsequently, it is also possible to multiply encrypted vectors with non-encrypted matrices, as it is often required for ML.

Since cryptography is a whole research field on its own, most FL papers using homomorphic encryption briefly explain the method, as well as the cryptography scheme being used, but there is no added research work done on homomorphic encryption for FL specifically. Most papers implement the Paillier [113] cryptosystem [13, 24, 45, 64] or the efficient PPDM [121] encryption [23]. A slightly different route is taken by Zhang et al. [76], who selected the ElGamal system [96] as a multiplicative homomorphic encryption scheme. This is because the authors opt for taking the exponential of weights, then to find the exponential of the aggregated weights of all clients one has to multiply the locally computed encrypted weights $[[\exp(\theta_k^t)]]$.

3.5.3 Secure Multi-Party Computation. The final method used for avoiding attacks on FL systems is secure multi-party computation. It describes a system of collaborating parties that share the goal of computing some value based on each of their private datasets. As such, it is similar to FL, however, multi-party computation systems are mostly very specialised

for the computation task at hand, whereas FL is focused on ML computation and strives to be applicable to a variety of datasets and tasks. Like aforementioned defence measures, multi-party computation also increases the computational effort for training an ML model.

One common examples of a multi-party computation task is *secret sharing* [120]. The scheme after Adi Shamir is used for splitting a secret amongst n parties, such that k or more parties together can reassemble the secret, but $k-1$ or fewer parties cannot. This works by setting up a polynomial function of degree $(k-1)$: $f = \sum_{i=0}^{n-1} a_i x^i$ with a_0 being the secret, and $\forall i \in [1, n-1]$, a_i are random numbers. Then each party receives a single point from this function, and k of those points can be used to compute the function equation and subsequently take the y-axis intersection as the secret. Secret sharing has been used for FL by Bonawitz et al. [6] and Liu et al. [46] to securely compute aggregate parameter values. In contrast, Zhang et al. [76] adopt this method to distribute a private encryption key to multiple clients.

3.5.4 Implications for Digital Health. All three aforementioned defence methods are applicable to healthcare datasets, however, none of the health-related papers in the review used them, except for two uses of homomorphic encryption [12, 39]. This is probably due to the missing reports of real-world applications which would require some combination of differential privacy, homomorphic encryption and multi-party computation. A limitation for especially encryption, but also the other concepts could be lacking computing power in hospitals. We will discuss this further in Section 4.2.1.

3.6 Health

In healthcare, data privacy is a crucial topic. With the introduction of the General Data Protection Regulation (GDPR) in May 2018, European patients' rights to their data has been increased even more, making it very challenging for research in this domain, especially across multiple hospitals. FL promises a privacy-preserving solution for this and there is already evidence of FL working well with openly accessible medical datasets, however, we did not find a paper in our literature review in which the paradigm has been deployed in the wild. Only 11 papers included in the review develop algorithms and methods for FL in digital health, all based on simulated data federation and without real-life deployment.

Special to the medical setting of FL is the fairly limited amount of clients (2-100), being hospitals or doctors, and the relatively high level of trust between them, which has been termed *cross-silo FL* by Kairouz et al. [103]. An exception from this would be an FL system, building on health-related data collected on people's smartphones. The following subsections will provide insight into the existing literature on FL for medical data, split by the type of data considered.

3.6.1 EHR/ICU/Genomics data. The first group of papers used electronic health records (EHRs) or data from intensive care units (ICUs). This is mainly tabular data describing patients' previous treatments, medication intake, etc. Also, genomic data has been used, but only by one paper included in the review [42]. As a common theme, most papers aim at predicting patient mortality, re-hospitalisation or patient discharge time.

The *VERTIGO* algorithm for vertically distributed data by Li et al. [42] was already mentioned and explained in Section 3.1.2. The tasks they wanted to solve, were binary prediction of mortality based on genome data for breast cancer patients, EHR data for myocardial infarction, or ICU data from the popular database *MIMIC-II* [86] (newer papers use the updated dataset *MIMIC-III*).

Other previously discussed papers are [31], where again the authors used their algorithm *LoAdaBoost* for predicting patient survival status based on the *MIMIC-III* database, or [61], which describes an optimal model selection process for different genome datasets (see Section 3.2.1).

In contrast to most other FL literature, Lee et al. [39] try to solve an unsupervised ML task in the form of a k-nearest neighbour (k-NN) model based on hashed EHRs. They consider different *data sources* like demographics, prescription data, lab tests or diagnoses, and use separate hash algorithms for each. Hashed data for patients can then be compared using the Hamming distance (number of differences in binary encoding) and finally, the algorithm is able to identify similar patients across different hospitals. This information can be used for clinical trial recruitment in multi-centre studies, or for disease surveillance across hospitals.

Similarly, Huang and Liu [30] perform patient clustering, but with the goal of training multiple more powerful and specialised NNs instead of one global one. Their algorithm, *community-based federated learning (CBFL)*, includes three steps. First, each hospital (50 included in the dataset, each containing 560 critical care patients) learns a denoising autoencoder model and shares the encoder weights with the server. All encoders are aggregated and the resulting model is sent back to each client. The second step is a k-means clustering algorithm, taking as input the average encoded features, it receives from each hospital using the previously trained global encoder model. Finally, k NN models are initialised and trained in parallel by all clients using the FL algorithm. One important distinction is that clients allocate each local data sample to one of the k clusters, and the count of samples in each cluster determines the factor, to which that client's weight updates influence the k-th NN. The final predictions made in this paper are patient mortality and hospital stay-time. An additional benefit it the possibility to analyse patient community distributions.

Liu et al. [44] suggest a Federated-Autonomous Deep Learning (FADL) approach, where after an initial FL phase, each participating client trains the deep layers of the neural network to optimise the model for local data. This follows the observation of Chen et al. [11] that shallow layers learn superordinate concepts applicable to a wider range of datasets, whereas deep layers are far more specialised on the data at hand. Liu et al. show that their approach outperforms classical FL and reaches an accuracy (measured by area under the receiver operating characteristic (AUROC)) comparable to centralised learning for the binary mortality prediction task based on ICU data from the eICU Collaborative Research Database [82].

In [7], Brisimi et al. solve another binary classification problem for predicting hospitalisations from EHRs, but in contrast to [44] the paper describes a federated framework for training a sparse SVM. The benefit of this model is the interpretability of the weight vector to detect features which high predictive value for future hospitalisation. Another considerable difference is the modelling of the client network without a server, in a fully decentralised manner.

3.6.2 Image data. Another common and important domain in digital health is medical image data. Here, it is especially difficult to preserve privacy while still using the data, since it is not clear how to anonymise images. They are considered identifiable if there is the possibility that someone recognises the patient by looking at the image. Recent work [88, 110] has dealt with the detection and removal of sensitive textual information in medical images following the common DICOM (Digital Imaging and Communications in Medicine) standard [81], but the image itself stays the same, leaving the problem of re-identification. Thus, FL can help in this area, keeping images directly in hospitals, but still allowing for large sets of training data for models.

Using a brain tumour segmentation dataset, Sheller et al. [58] implemented the FL algorithm for CNN image segmentation. In comparison to other collaborative learning approaches, FL reached the highest accuracy (99% of the performance for data-sharing and central model training) and can scale better with the number of collaborating institutions.

3.6.3 Sensor data. With an increasing focus on IoHT research and wearable technology, sensor data has to be analysed quickly and reliably using ML models.

A previously discussed method for a HAR dataset can be found in [60], where excluding bad clients in federated regression models and NNs is evaluated for this task (see Section 3.2). Also, Sanyal et al. [56] investigated a federated filtering framework for a public multivariate, time-series IoHT dataset of patients performing 12 physical activities (MHEALTH [85]). They simulated a Least Mean Square (LMS) filter [101] on each device and used a fog server to combine the individual prediction models to estimate a perturbed data matrix (under protection of data privacy), update local filter parameters and perform global decision making. The authors reported a very low communication effort and high scalability of their approach.

The other sensor-data related paper included in this review is [12], in which a HAR task is solved using inertial measurement unit (IMU) data (accelerometer, gyroscope) from smartphones. In contrast to the previously mentioned paper, Chen et al. adopt a CNN for prediction and used federated transfer learning to personalise the model for individual people. For the transfer learning approach `FedHealth`, local model training includes a *correlation alignment* loss term, considering the global model. The average prediction accuracy across the five people in the dataset lies at 99.4%, which is more than 5pp higher than the baseline federated models.

4 DISCUSSION

The goal of this systematic literature review was to investigate the following two research questions:

RQ1: What is the state of the art in the field of FL, and what are its limitations?
RQ2: Which areas of FL research are most promising for digital health applications?

A systematic evaluation of all included papers is complex, because a multitude of participant settings and datasets were used. In order to make comparisons between proposed algorithms, authors should include benchmarking datasets such as the ones listed by *LEAF* [84].

The previous chapter provided an extensive overview of the existing FL approaches, which was subject of RQ1 (see Section 2.1). Clearly, the field of FL has been growing a lot since 2016, and the amount of literature will most likely continue to grow as more researchers adopt the federated approach to learning, due to its benefits for data privacy. On the other hand, if it is possible to consolidate data and learn a centralised model, there are still many benefits of going that route.

4.1 Open Questions

We determined the following limitations and open questions for FL.

4.1.1 Unsupervised machine learning. Much literature has looked into supervised FL approaches, unsupervised ML, on the other hand, has been mostly overlooked so far. There are only 2 papers included in this review, who considered a k-NN model, and we believe that this is a much-needed research direction, especially in cases where data labels might be hard to come by, like in medicine.

4.1.2 Hyperparameter optimisation. Another limiting factor is the requirement of a pre-defined model structure. Usually, in the ML development cycle, one of the first steps is to select a proper model type and optimise the hyperparameters. This is difficult to achieve for a federated dataset since clients would have to contribute to hyperparameter optimisation before actually getting a benefit out of their involvement in the FL system. Moreover, if the system includes differential privacy, there is a limited privacy budget that will already be spent on selecting a suitable model, and data privacy cannot be guaranteed for model training.

An easy parameter to update on the fly is the learning rate, and there has been research into updating it adaptively [38]. An approach for updating the overall NN model structure is using an evolutionary algorithm, which is highly computationally expensive since a population of possible ML models are trained simultaneously [79]. Especially low-battery, low-resource Internet of Things (IoT) devices or even smartphones will not be able to participate in such a system. One could also choose to follow a transfer learning approach, looking for models, that have performed well on a particular task before, and training those in a federated manner [12, 45].

4.2 Federated Learning for Digital Health

Looking at the state of FL for healthcare, there are only 11 papers included in the review, which apply their algorithms to medical data. On the other hand, 33 papers, so almost half, identify healthcare as an area that can benefit a lot from adopting privacy-preserving and distributed ML.

We find that vertical FL can be incredibly useful in the medical field, in order to get a more complete picture of patients, their visits to different doctors, and the corresponding data that was collected. Vertical FL can be applied to data collected by wearables as well as smartphones and smart sensing homes could in the future combine sensed data for prediction of patterns and diseases.

4.2.1 Privacy & Security. The goal of ML for healthcare should be to use the trained models as a trustable advisor to healthcare professionals. In some cases, an incorrect model prediction could advise the wrong treatment, or it could not detect a person which could be at risk. This could potentially have fatal consequences, which makes the security of an FL system very important. Before using actual FL systems in hospitals, there needs to be a guarantee that no adversary can breach it using e.g. model poisoning.

Out of the 11 papers concerning health-related data, two [12, 39] use homomorphic encryption, to hinder attacks like reverse model engineering. None of them included differential privacy, although Sheller et al. [58] acknowledged the potential benefits of differential privacy for their image segmentation model, but leave it to future work. This stands in contrast to the fact, that medical data is incredibly sensitive and FL models for it should implement all possible defensive measures. We identify a need for more research in that direction, making it possible to deploy the systems in real hospital and IoHT environments.

4.2.2 Limitations. One restriction that applies especially in the healthcare field is the requirement of the same data format for horizontal FL. Hospitals and clinics might use very different ways of collecting their data, and there is a huge amount of unstructured and textual data, that is not easily usable for FL. EHR data standards like *Fast Healthcare Interoperability Resources (FHIR)* [83] help alleviate this, but especially sensor data will be widely different. Subsequently, FL for health often requires some amount of data preprocessing on the side of the medical partner (since researchers are not allowed to access the private data directly). One can imagine providing hospitals with a data format template, that they have to conform with before starting the training process.

Another limitation for deploying FL systems in hospitals and doctor's practices is the computational resource requirement of training ML models. Oftentimes, the computer equipment in hospitals is not meant for gradient computation and lacks GPU power. Other than for IoHT applications, this only slows down the progress, but FL is still possible, since there is no need for fast model training, and it is okay if the computation takes longer to come up with a good model. IoHT may require applying some of the concepts from Section 3.2.4 in order to deal with low battery and compute power.

Finally, there is the limitation of finding appropriate medical institutions, willing to contribute to an FL system, but also equipped with the IT infrastructure to enable a pipeline from live data connection to the model, training and inference to transmitting new model parameters via secure channels to the aggregating server.

Overall, we find that the papers on FL for digital health describe a very heterogeneous set of methods that make a systematic evaluation difficult. There exist no standards for FL systems in this field which is something research should strive towards, as more papers on this topic get published.

4.2.3 Possible Future Research Directions. In addition to the privacy and security aspect of FL for healthcare, the healthcare applications considered in this review are far from exhausted. We require more research into using EHR data, because of the benefits of using a predefined data standard for horizontal FL. This data can lead to new models for clinical decision making and better risk modelling or can be used to find patients for a specific clinical trial [39].

With the large amount of text data in doctor's letters and medical reports, there is furthermore a need for federated natural language processing (NLP) models that can make use of it. The challenge here is to adequately embed features from those documents in order to use them as input for ML systems and improve their power using multiple data sources in an FL system.

We believe that another opportunity of FL for healthcare is the area of genomics. Aside from the fact, that anonymisation of genetic data is not quite clear under *GDPR* regulation and re-identification may be possible [106], the size of single data samples of this type limits sharing it on a large scale. Moreover, most existing datasets are quite small, meaning there are fewer samples than features which is challenging for ML. Being able to combine what little data exists in medical institutions could go a long way towards detecting associations between genotypes and diseases.

An area that can be explored well with FL is making use of data collected by patients at home using wearable medical devices or the patients' phones. Predicting the necessity of an intervention in the context of remote patient monitoring could rely solely on privately collected and processed data, while a smartphone app could provide an interface for patients to label the data by logging their current condition. We believe that not only physical health, but also mental health can benefit from the FL principle and sensitive data like social network or smartphone usage in the future.

5 CONCLUSION

We showed the progress of FL over the last 4 years in terms of training algorithms, security and privacy protocols, as well as communication efficiency and put them in context of an application in healthcare. We hope that this paper motivates more research into FL in general and healthcare applications, and we believe that in the future FL will become a standard for dealing with sensitive medical data. There are a number of open challenges for researchers, which include privacy-preserving hyperparameter optimisation, entity resolution for vertically split data and efficient ways of using encryption. We expect that FL for medical purposes will see increased popularity in the near future which should entail more sophisticated security and privacy guarantees allowing for real-world deployment of FL systems. Compared with other domains, the healthcare sector is in dire need for the potential advances made possible by ML and more specifically FL.

ACKNOWLEDGMENTS

This research was partly funded by the Federal Ministry of Education and Research of Germany in the framework of KI-LAB-ITSE (project number 01IS19066).
This research was partly funded by the HPI Research School on Data Science and Engineering.

REVIEWED LITERATURE

[1] Mohammad Mohammadi Amiri and Deniz Gündüz. 2019. Federated Learning over Wireless Fading Channels. *CoRR* abs/1907.09769 (2019). arXiv:1907.09769 http://arxiv.org/abs/1907.09769

[2] Muhammad Ammad-ud-din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System. *CoRR* abs/1901.09888 (2019). arXiv:1901.09888 http://arxiv.org/abs/1901.09888

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2018. How To Backdoor Federated Learning. *CoRR* abs/1807.00459 (2018). arXiv:1807.00459 http://arxiv.org/abs/1807.00459

[4] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing Federated Learning through an Adversarial Lens. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 634–643. http://proceedings.mlr.press/v97/bhagoji19a.html

[5] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. 2018. Protection Against Reconstruction and Its Applications in Private Federated Learning. *arXiv preprint arXiv:1812.00984* (2018).

[6] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. ACM, New York, NY, USA, 1175–1191. https://doi.org/10.1145/3133956.3133982

[7] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi. 2018. Federated learning of predictive models from federated Electronic Health Records. *Int J Med Inform* 112 (04 2018), 59–67.

[8] Sebastian Caldas, Jakub Konecný, H. Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *CoRR* abs/1812.07210 (2018). arXiv:1812.07210 http://arxiv.org/abs/1812.07210

[9] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2019. Secure Federated Matrix Factorization. *CoRR* abs/1906.05108 (2019). arXiv:1906.05108 http://arxiv.org/abs/1906.05108

[10] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated Meta-Learning for Recommendation. *CoRR* abs/1802.07876 (2018). arXiv:1802.07876 http://arxiv.org/abs/1802.07876

[11] Yang Chen, Xiaoyan Sun, and Yaochu Jin. 2019. Communication-Efficient Federated Deep Learning with Asynchronous Model Update and Temporally Weighted Aggregation. *CoRR* abs/1903.07424 (2019). arXiv:1903.07424 http://arxiv.org/abs/1903.07424

[12] Yiqiang Chen, Jindong Wang, Chaohui Yu, Wen Gao, and Xin Qin. 2019. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. *CoRR* abs/1907.09173 (2019). arXiv:1907.09173 http://arxiv.org/abs/1907.09173

[13] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. 2019. SecureBoost: A Lossless Federated Learning Framework. *CoRR* abs/1901.08755 (2019). arXiv:1901.08755 http://arxiv.org/abs/1901.08755

[14] Luca Corinzia and Joachim M. Buhmann. 2019. Variational Federated Multi-Task Learning. *CoRR* abs/1906.06268 (2019). arXiv:1906.06268 http://arxiv.org/abs/1906.06268

[15] Moming Duan. 2019. Astraea: Self-balancing Federated Learning for Improving Classification Accuracy of Mobile Deep Learning Applications. *CoRR* abs/1907.01132 (2019). arXiv:1907.01132 http://arxiv.org/abs/1907.01132

[16] Hubert Eichner, Tomer Koren, H. Brendan McMahan, Nathan Srebro, and Kunal Talwar. 2019. Semi-Cyclic Stochastic Gradient Descent. *CoRR* abs/1904.10120 (2019). arXiv:1904.10120 http://arxiv.org/abs/1904.10120

[17] Shaohan Feng, Dusit Niyato, Ping Wang, Dong In Kim, and Ying-Chang Liang. 2018. Joint Service Pricing and Cooperative Relay Communication for Federated Learning. *CoRR* abs/1811.12082 (2018). arXiv:1811.12082 http://arxiv.org/abs/1811.12082

[18] Clement Fung, Jamie Koerner, Stewart Grant, and Ivan Beschastnikh. 2018. Dancing in the Dark: Private Multi-Party Machine Learning in an Untrusted Setting. *CoRR* abs/1811.09712 (2018). arXiv:1811.09712 http://arxiv.org/abs/1811.09712

[19] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2018. Mitigating Sybils in Federated Learning Poisoning. *CoRR* abs/1808.04866 (2018). arXiv:1808.04866 http://arxiv.org/abs/1808.04866

[20] S. Gade and N. H. Vaidya. 2018. Privacy-Preserving Distributed Learning via Obfuscated Stochastic Gradients. In *2018 IEEE Conference on Decision and Control (CDC)*. 184–191. https://doi.org/10.1109/CDC.2018.8619133

[21] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially Private Federated Learning: A Client Level Perspective. *CoRR* abs/1712.07557 (2017). arXiv:1712.07557 http://arxiv.org/abs/1712.07557

[22] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-Shot Federated Learning. *CoRR* abs/1902.11175 (2019). arXiv:1902.11175 http://arxiv.org/abs/1902.11175

[23] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang. 2019. Towards Efficient and Privacy-Preserving Federated Deep Learning. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC.2019.8761267

[24] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *CoRR* abs/1711.10677 (2017). arXiv:1711.10677 http://arxiv.org/abs/1711.10677

[25] Valentin Hartmann, Konark Modi, Josep M. Pujol, and Robert West. 2019. Privacy-Preserving Classification with Secret Vector Machines. *CoRR* abs/1907.03373 (2019). arXiv:1907.03373 http://arxiv.org/abs/1907.03373

[26] X. He, Q. Ling, and T. Chen. 2019. Byzantine-Robust Stochastic Gradient Descent for Distributed Low-Rank Matrix Completion. In *2019 IEEE Data Science Workshop (DSW)*. 322–326. https://doi.org/10.1109/DSW.2019.8755575

[27] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. *CoRR* abs/1702.07464 (2017). arXiv:1702.07464 http://arxiv.org/abs/1702.07464

[28] B. Hu, Y. Gao, L. Liu, and H. Ma. 2018. Federated Region-Learning: An Edge Computing Based Framework for Urban Environment Sensing. In *2018 IEEE Global Communications Conference (GLOBECOM)*. 1–7. https://doi.org/10.1109/GLOCOM.2018.8647649

[29] Yao Hu, Xiaoyan Sun, Yang Chen, and Zishuai Lu. 2019. Model and Feature Aggregation Based Federated Learning for Multi-sensor Time Series Trend Following. In *Advances in Computational Intelligence*, Ignacio Rojas, Gonzalo Joya, and Andreu Catala (Eds.). Springer International Publishing, Cham, 233–246.

[30] Li Huang and Dianbo Liu. 2019. Patient Clustering Improves Efficiency of Federated Machine Learning to predict mortality and hospital stay time using distributed Electronic Medical Records. *CoRR* abs/1903.09296 (2019). arXiv:1903.09296 http://arxiv.org/abs/1903.09296

[31] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. 2018. LoAdaBoost: Loss-Based AdaBoost Federated Machine Learning on medical Data. *CoRR* abs/1811.12629 (2018). arXiv:1811.12629 http://arxiv.org/abs/1811.12629

[32] Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. 2018. Learning Private Neural Language Modeling with Attentive Aggregation. *CoRR* abs/1812.07108 (2018). arXiv:1812.07108 http://arxiv.org/abs/1812.07108

[33] Eugene Kharitonov. 2019. Federated Online Learning to Rank with Evolution Strategies. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. ACM, New York, NY, USA, 249–257. https://doi.org/10.1145/3289600.3290968

[34] Jakub Konecný, Brendan McMahan, and Daniel Ramage. 2015. Federated Optimization: Distributed Optimization Beyond the Datacenter. *CoRR* abs/1511.03575 (2015). arXiv:1511.03575 http://arxiv.org/abs/1511.03575

[35] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR* abs/1610.02527 (2016). arXiv:1610.02527 http://arxiv.org/abs/1610.02527

[36] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR* abs/1610.05492 (2016). arXiv:1610.05492 http://arxiv.org/abs/1610.05492

[37] Jakub Konecný and Peter Richtárik. 2016. Randomized Distributed Mean Estimation: Accuracy vs Communication. *CoRR* abs/1611.07555 (2016). arXiv:1611.07555 http://arxiv.org/abs/1611.07555

[38] Antti Koskela and Antti Honkela. 2018. Learning rate adaptation for federated and differentially private learning descent. *CoRR* abs/1809.03832 (2018). arXiv:1809.03832 http://arxiv.org/abs/1809.03832

[39] Junghye Lee, Jimeng Sun, Fei Wang, Shuang Wang, Chi-Hyuck Jun, and Xiaoqian Jiang. 2018. Privacy-Preserving Patient Similarity Learning in a Federated Environment: Development and Analysis. *JMIR Med Inform* 6, 2 (13 Apr 2018), e20. https://doi.org/10.2196/medinform.7744

[40] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. 2019. Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6341–6345.

[41] Tian Li, Maziar Sanjabi, and Virginia Smith. 2019. Fair Resource Allocation in Federated Learning. *CoRR* abs/1905.10497 (2019). arXiv:1905.10497 http://arxiv.org/abs/1905.10497

[42] Y. Li, X. Jiang, S. Wang, H. Xiong, and L. Ohno-Machado. 2016. VERTIcal Grid lOgistic regression (VERTIGO). *J Am Med Inform Assoc* 23, 3 (05 2016), 570–579.

[43] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. 2017. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. *CoRR* abs/1712.01887 (2017). arXiv:1712.01887 http://arxiv.org/abs/1712.01887

[44] Dianbo Liu, Timothy Miller, Raheel Sayeed, and Kenneth D. Mandl. 2018. FADL: Federated-Autonomous Deep Learning for Distributed Electronic Health Record. *CoRR* abs/1811.11400 (2018). arXiv:1811.11400 http://arxiv.org/abs/1811.11400

[45] Yang Liu, Tianjian Chen, and Qiang Yang. 2018. Secure Federated Transfer Learning. *CoRR* abs/1812.03337 (2018). arXiv:1812.03337 http://arxiv.org/abs/1812.03337

[46] Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, Surya Nepal, and Robert H. Deng. 2019. Boosting Privately: Privacy-Preserving Federated Extreme Boosting for Mobile Crowdsensing. *CoRR* abs/1907.10218 (2019). arXiv:1907.10218 http://arxiv.org/abs/1907.10218

[47] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated Learning of Deep Networks using Model Averaging. *CoRR* abs/1602.05629 (2016). arXiv:1602.05629 http://arxiv.org/abs/1602.05629

[48] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. Learning Differentially Private Language Models Without Losing Accuracy. *CoRR* abs/1710.06963 (2017). arXiv:1710.06963 http://arxiv.org/abs/1710.06963

[49] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE Symposium on Security and Privacy, 2019*. 691–706. https://doi.org/10.1109/SP.2019.00029

[50] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic Federated Learning. *CoRR* abs/1902.00146 (2019). arXiv:1902.00146 http://arxiv.org/abs/1902.00146

[51] M. Nasr, R. Shokri, and A. Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 2019 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 1021–1035. https://doi.org/10.1109/SP.2019.00065

[52] Takayuki Nishio and Ryo Yonetani. 2018. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. *CoRR* abs/1804.08333 (2018). arXiv:1804.08333 http://arxiv.org/abs/1804.08333

[53] Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2018. Entity Resolution and Federated Learning get a Federated Resolution. *CoRR* abs/1803.04035 (2018). arXiv:1803.04035 http://arxiv.org/abs/1803.04035

[54] Tribhuvanesh Orekondy, Seong Joon Oh, Bernt Schiele, and Mario Fritz. 2018. Understanding and Controlling User Linkability in Decentralized Learning. *CoRR* abs/1805.05838 (2018). arXiv:1805.05838 http://arxiv.org/abs/1805.05838

[55] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. 2018. Federated Optimization for Heterogeneous Networks. *CoRR* abs/1812.06127 (2018). arXiv:1812.06127 http://arxiv.org/abs/1812.06127

[56] Sunny Sanyal, Dapeng Wu, and Boubakr Nour. 2019. A Federated Filtering Framework for Internet of Medical Things. *CoRR* abs/1905.01138 (2019). arXiv:1905.01138 http://arxiv.org/abs/1905.01138

[57] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and Communication-Efficient Federated Learning from Non-IID Data. *CoRR* abs/1903.02891 (2019). arXiv:1903.02891 http://arxiv.org/abs/1903.02891

[58] Micah J. Sheller, G. Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. 2019. Multi-institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Alessandro Crimi, Spyridon Bakas, Hugo Kuijf, Farahani Keyvan, Mauricio Reyes, and Theo van Walsum (Eds.). Springer International Publishing, Cham, 92–104.

[59] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated Multi-Task Learning. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4424–4434. http://papers.nips.cc/paper/7029-federated-multi-task-learning.pdf

[60] K. Sozinov, V. Vlassov, and S. Girdzijauskas. 2018. 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)*. 1103–1111. https://doi.org/10.1109/BDCloud.2018.00164

[61] Xudong Sun, Andrea Bommert, Florian Pfisterer, Jörg Rahnenführer, Michel Lang, and Bernd Bischl. 2019. High Dimensional Restrictive Federated Model Selection with multi-objective Bayesian Optimization over shifted distributions. *CoRR* abs/1902.08999 (2019). arXiv:1902.08999 http://arxiv.org/abs/1902.08999

[62] Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. 2017. Distributed Mean Estimation with Limited Communication. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, International Convention Centre, Sydney, Australia, 3329–3337. http://proceedings.mlr.press/v70/suresh17a.html

[63] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong. 2019. Federated Learning over Wireless Networks: Optimization Model Design and Analysis. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 1387–1395. https://doi.org/10.1109/INFOCOM.2019.8737464

[64] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, and Rui Zhang. 2018. A Hybrid Approach to Privacy-Preserving Federated Learning. *CoRR* abs/1812.03224 (2018). arXiv:1812.03224 http://arxiv.org/abs/1812.03224

[65] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. 2018. Split learning for health: Distributed deep learning without sharing raw patient data. *CoRR* abs/1812.00564 (2018). arXiv:1812.00564 http://arxiv.org/abs/1812.00564

[66] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu. 2018. Deep Learning towards Mobile Applications. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 1385–1393. https://doi.org/10.1109/ICDCS.2018.00139

[67] L. Wang, W. Wang, and B. LI. 2019. CMFL: Mitigating Communication Overhead for Federated Learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 954–964. https://doi.org/10.1109/ICDCS.2019.00099

[68] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2018. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *CoRR* abs/1804.05271 (2018). arXiv:1804.05271 http://arxiv.org/abs/1804.05271

[69] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2018. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. *CoRR* abs/1812.00535 (2018). arXiv:1812.00535 http://arxiv.org/abs/1812.00535

[70] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Asynchronous Federated Optimization. *CoRR* abs/1903.03934 (2019). arXiv:1903.03934 http://arxiv.org/abs/1903.03934

[71] Kele Xu, Haibo Mi, Dawei Feng, Huaimin Wang, Chuan Chen, Zibin Zheng, and Xu Lan. 2018. Collaborative Deep Learning Across Multiple Data Centers. *CoRR* abs/1810.06877 (2018). arXiv:1810.06877 http://arxiv.org/abs/1810.06877

[72] Zichen Xu, Li Li, and Wenting Zou. 2019. Exploring Federated Learning on Battery-Powered Devices. In *Proceedings of the ACM Turing Celebration Conference - China* (Chengdu, China) *(ACM TURC '19)*. Association for Computing Machinery, New York, NY, USA, Article 6, 6 pages. https://doi.org/10.1145/3321408.3323080

[73] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. 2018. Federated Learning via Over-the-Air Computation. *CoRR* abs/1812.11750 (2018). arXiv:1812.11750 http://arxiv.org/abs/1812.11750

[74] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12 (Jan. 2019), 19 pages. https://doi.org/10.1145/3298981

[75] X. Yao, C. Huang, and L. Sun. 2018. Two-Stream Federated Learning: Reduce the Communication Costs. In *2018 IEEE Visual Communications and Image Processing (VCIP)*. 1–4. https://doi.org/10.1109/VCIP.2018.8698609

[76] X. Zhang, S. Ji, H. Wang, and T. Wang. 2017. Private, Yet Practical, Multiparty Deep Learning. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 1442–1452. https://doi.org/10.1109/ICDCS.2017.215

[77] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. *CoRR* abs/1806.00582 (2018). arXiv:1806.00582 http://arxiv.org/abs/1806.00582

[78] W. Zhou, Y. Li, S. Chen, and B. Ding. 2018. Real-Time Data Processing Architecture for Multi-Robots Based on Differential Federated Learning. In *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 462–471. https://doi.org/10.1109/SmartWorld.2018.00106

[79] Hangyu Zhu and Yaochu Jin. 2018. Multi-objective Evolutionary Federated Learning. *CoRR* abs/1812.07478 (2018). arXiv:1812.07478 http://arxiv.org/abs/1812.07478

[80] Y. Zou, S. Feng, D. Niyato, Y. Jiao, S. Gong, and W. Cheng. 2019. Mobile Device Training Strategies in Federated Learning: An Evolutionary Game Approach. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 874–879. https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00157

ADDITIONAL REFERENCES

[81] [n.d.]. *DICOM Standard.* Retrieved January 25, 2020 from https://www.dicomstandard.org/

[82] [n.d.]. *eICU Collaborative Research Database.* Retrieved January 25, 2020 from https://eicu-crd.mit.edu/

[83] [n.d.]. *Index - FHIR v4.0.1.* Retrieved January 27, 2020 from https://www.hl7.org/fhir/

[84] [n.d.]. *LEAF.* Retrieved January 27, 2020 from https://leaf.cmu.edu/

[85] [n.d.]. *MHEALTH Data Set.* Retrieved January 25, 2020 from https://archive.ics.uci.edu/ml/datasets/MHEALTH+Dataset

[86] [n.d.]. *MIMIC Critical Care Database.* Retrieved January 25, 2020 from https://mimic.physionet.org/

[87] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. ACM, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[88] K Y E Aryanto, M Oudkerk, and P M A van Ooijen. 2015. Free DICOM de-identification tools in clinical research: functioning and safety of patient privacy. *European radiology* 25, 12 (12 2015), 3685–3695.

[89] Kitchenham BA and Stuart Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. 2 (01 2007).

[90] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. 2010. The security of machine learning. *Machine Learning* 81, 2 (2010), 121–148.

[91] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning Attacks Against Support Vector Machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning* (Edinburgh, Scotland) *(ICML'12)*. Omnipress, USA, 1467–1474. http://dl.acm.org/citation.cfm?id=3042573.3042761

[92] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317 – 331. https://doi.org/10.1016/j.patcog.2018.07.023

[93] John S. Breese, David Heckerman, and Carl Kadie. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (Madison, Wisconsin) *(UAI'98)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 43–52. http://dl.acm.org/citation.cfm?id=2074094.2074100

[94] Chris Culnane, Benjamin I. P. Rubinstein, and Vanessa Teague. 2017. Health Data in an Open World. *CoRR* abs/1712.05627 (2017). arXiv:1712.05627 http://arxiv.org/abs/1712.05627

[95] John R. Douceur. 2002. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS '01)*. Springer-Verlag, Berlin, Heidelberg, 251–260.

[96] T. Elgamal. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory* 31, 4 (July 1985), 469–472. https://doi.org/10.1109/TIT.1985.1057074

[97] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1322–1333. https://doi.org/10.1145/2810103.2813677

[98] Milena A Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. 2018. Potential Biases in Machine Learning Algorithms Using Electronic Health Record Data. *JAMA internal medicine* 178, 11 (11 2018), 1544–1547.

[99] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[100] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML]

[101] S. Haykin and B. Widrow. 2003. *Least-Mean-Square Adaptive Filters.* Wiley. https://books.google.de/books?id=U8X3mJtawUkC

[102] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. *CoRR* abs/1804.00308 (2018). arXiv:1804.00308 http://arxiv.org/abs/1804.00308

[103] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón,

Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2019. Advances and Open Problems in Federated Learning. arXiv:1912.04977 [cs.LG]

[104] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images.* Technical Report. Citeseer.

[105] Yann LeCun, Y. Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521 (05 2015), 436–44. https://doi.org/10.1038/nature14539

[106] Bradley Malin and Latanya Sweeney. 2004. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics* 37, 3 (2004), 179 – 192. https://doi.org/10.1016/j.jbi.2004.04.005

[107] Frank D. McSherry. 2009. Privacy Integrated Queries: An Extensible Platform for Privacy-preserving Data Analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data* (Providence, Rhode Island, USA) *(SIGMOD '09)*. ACM, New York, NY, USA, 19–30. https://doi.org/10.1145/1559845.1559850

[108] Xianrui Meng, Dimitrios Papadopoulos, Alina Oprea, and Nikos Triandopoulos. 2019. Privacy-Preserving Hierarchical Clustering: Formal Security and Efficient Approximation. *CoRR* abs/1904.04475 (2019). arXiv:1904.04475 http://arxiv.org/abs/1904.04475

[109] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, and The PRISMA Group. 2009. Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. *PLOS Medicine* 6, 7 (07 2009), 1–6. https://doi.org/10.1371/journal.pmed.1000097

[110] E. Monteiro, C. Costa, and J. L. Oliveira. 2015. A machine learning methodology for medical imaging anonymization. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 1381–1384. https://doi.org/10.1109/EMBC.2015.7318626

[111] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2015. DeepFool: a simple and accurate method to fool deep neural networks. *CoRR* abs/1511.04599 (2015). arXiv:1511.04599 http://arxiv.org/abs/1511.04599

[112] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453.

[113] Pascal Paillier. 1999. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques* (Prague, Czech Republic) *(EUROCRYPT'99)*. Springer-Verlag, Berlin, Heidelberg, 223–238. http://dl.acm.org/citation.cfm?id=1756123.1756146

[114] Diego Peteiro-Barral and Bertha Guijarro-Berdiñas. 2013. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence* 2, 1 (2013), 1–11. https://doi.org/10.1007/s13748-012-0035-5

[115] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. 2015. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 896–902.

[116] R L Rivest, L Adleman, and M L Dertouzos. 1978. On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation, Academia Press* (1978), 169–179.

[117] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre de Montjoye. 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications* 10, 1 (2019), 3069.

[118] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.

[119] Z. Shae and J. Tsai. 2018. Transform Blockchain into Distributed Parallel Computing Architecture for Precision Medicine. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 1290–1299. https://doi.org/10.1109/ICDCS.2018.00129

[120] Adi Shamir. 1979. How to Share a Secret. *Commun. ACM* 22, 11 (Nov. 1979), 612–613. https://doi.org/10.1145/359168.359176

[121] J. Zhou, Z. Cao, X. Dong, and X. Lin. 2015. PPDM: A Privacy-Preserving Protocol for Cloud-Assisted e-Healthcare Systems. *IEEE Journal of Selected Topics in Signal Processing* 9, 7 (Oct 2015), 1332–1344. https://doi.org/10.1109/JSTSP.2015.2427113

A ACRONYMS

CNN . . . convolutional neural network
EHR . . . electronic health record
EMD . . . earth mover's distance
FL . . . . federated learning
GAN . . . generative adversarial net
GRU . . . gated recurrent unit
HAR . . . human activity recognition
ICU . . . . intensive care unit
IID . . . . independent and identically distributed
IMU . . . . inertial measurement unit
IoHT . . . Internet of Health Things
IoT . . . . Internet of Things
k-NN . . . k-nearest neighbour
LSTM . . . long short-term memory
ML . . . . machine learning
MSE . . . mean squared error
NLP . . . . natural language processing
NN . . . . neural network
RNN . . . recurrent neural network
SGD . . . stochastic gradient descent
SVM . . . support vector machine

B DATA COLLECTION

Table 2. Data collected from papers included in the review. Abbreviations:
Ref.: Reference; Alg.: Algorithm; S/P: Security/Privacy; Comm.: Communication Efficiency;
DP: Differential Privacy; MPC: Multi-Party Computation; HE: Homomorphic Encryption

Ref.	Year	Alg.	S./P.	Comm.	DP	MPC	HE	ML Models	Data
[1]	2019	0	0	1	0	0	0	NN	MNIST
[2]	2019	1	0	0	0	0	0	Collaborative Filter	In-House data, MovieLens, Synthetic data
[3]	2018	0	1	0	1	0	0	CNN, long short-term memory (LSTM)	CIFAR-10, Reddit comments
[4]	2019	0	1	0	0	0	0	CNN, NN	Adult Census Income, Fashion-MNIST

Table 2. (continued) Abbreviations:
Ref.: Reference; Alg.: Algorithm; S/P: Security/Privacy; Comm.: Communication Efficiency;
DP: Differential Privacy; MPC: Multi-Party Computation; HE: Homomorphic Encryption

Ref.	Year	Alg.	S./P.	Comm.	DP	MPC	HE	ML Models	Data
[5]	2018	0	1	0	0	0	0	CNN, LSTM	CIFAR-10, MNIST, Reddit comments, YFCC100M
[6]	2017	0	1	1	0	1	1	-	-
[7]	2018	1	0	1	0	0	0	sparse SVM (sSVM)	Boston Medical Center EHR
[8]	2019	1	0	1	0	0	0	CNN	CIFAR-10, EMNIST, MNIST
[9]	2019	1	1	0	0	0	1	Matrix Factorisation	MovieLens
[10]	2018	1	0	0	0	0	0	Logistic Regression, NN	Mobile Service Usage Records production dataset, MovieLens
[11]	2019	1	0	1	0	0	0	CNN, LSTM	HAR (not further defined), MNIST
[12]	2019	1	0	0	0	0	1	CNN	UCI Smartphone HAR
[13]	2019	1	1	1	0	0	1	tree-boosting system	Kaggle's "Give me some credit", Kaggle's "Default of Credit Card Clients"
[14]	2019	1	0	0	0	0	0	NN	FEMNIST, MNIST, P-MNIST, UCI Smartphone HAR, Vehicle Sensors Network,
[15]	2019	1	0	0	0	0	0	CNN	EMNIST
[16]	2019	1	0	0	0	0	0	Logistic Regression	Twitter Sentiment140
[17]	2018	0	0	1	0	0	0	-	-
[18]	2018	0	1	0	1	0	0	Logistic Regression	UCI credit card
[19]	2018	0	1	0	0	1	0	Logistic Regression (Multi-Class)	Amazon Reviews, KDDCub99, MNIST
[20]	2018	1	1	0	0	0	0	Linear Regression	Synthetic data
[21]	2017	0	1	0	1	0	0	NN	MNIST

Table 2. (continued) Abbreviations:
Ref.: Reference; Alg.: Algorithm; S/P: Security/Privacy; Comm.: Communication Efficiency;
DP: Differential Privacy; MPC: Multi-Party Computation; HE: Homomorphic Encryption

Ref.	Year	Alg.	S./P.	Comm.	DP	MPC	HE	ML Models	Data
[22]	2019	1	0	1	0	0	0	SVM	EMNIST, Google Glass (GLEAM), Twitter Sentiment140
[23]	2019	0	1	1	1	0	1	CNN	MNIST
[24]	2017	1	1	0	0	1	1	Logistic Regression	UCI Boston Housing, UCI Breast Cancer, UCI Diabetes, Kaggle's "Give me some credit", MNIST
[25]	2019	1	1	0	0	0	0	linear SVM with Integer features	Predict gender from Tweets, predict gender from websites visited in browser
[26]	2019	1	1	0	0	0	0	Matrix Factorisation	Kaggle's "Netflix Prize data"
[27]	2017	0	1	0	1	0	0	CNN	AT&T Faces, MNIST
[28]	2018	1	0	0	0	0	0	LSTM	Beijing weather & air pollution data
[29]	2019	1	0	0	0	0	0	LSTM	Real-World sensor data
[30]	2019	1	0	0	0	0	0	K-Means	eICU Collaborative Research Database
[31]	2018	1	0	0	0	0	0	NN	MIMIC-III
[32]	2018	1	0	0	1	0	0	gated recurrent unit (GRU)	Penn Treebank, Reddit comments, WikiText-2
[33]	2019	1	0	0	1	0	0	Linear Regression Ranking Model, NN	MQ2007, MQ2008
[34]	2015	1	0	1	0	0	0	Logistic Regression	Public Social Network Posts
[35]	2016	1	0	0	0	0	0	Logistic Regression	Public Google+ posts
[36]	2017	0	0	1	0	0	0	CNN, LSTM	CIFAR-10, Reddit comments
[37]	2016	1	0	1	0	0	0	-	Synthetic data

Table 2. (continued) Abbreviations:
Ref.: Reference; Alg.: Algorithm; S/P: Security/Privacy; Comm.: Communication Efficiency;
DP: Differential Privacy; MPC: Multi-Party Computation; HE: Homomorphic Encryption

Ref.	Year	Alg.	S./P.	Comm.	DP	MPC	HE	ML Models	Data
[38]	2019	1	1	0	1	0	0	CNN	CIFAR-10
[39]	2018	1	0	0	0	0	1	k-NN	MIMIC-III
[40]	2019	1	0	0	0	0	0	CNN	Crowdsourced "Hey Snips" Wakeword detection dataset
[41]	2019	1	0	0	0	0	0	Logistic Regression, linear SVM, stacked LSTM	Adult Census Income, FMNIST, Synthetic data, Twitter Sentiment140, Works of Shakespeare, Vehicle Sensors Network
[42]	2015	1	0	0	0	0	0	Logistic Regression	Genome (GSE3494), MIMIC-II, Myocardial infarction data, Synthetic data
[43]	2018	0	0	1	0	0	0	CNN, LSTM	CIFAR-10, Librispeech Corpus, Penn Treebank
[44]	2018	1	0	0	0	0	0	NN	eICU Collaborative Research Database
[45]	2018	1	1	0	0	1	1	NN (Stacked Autoencoder)	Kaggle's "Default of Credit Card Clients", NUS-WIDE
[46]	2019	1	1	0	0	1	0	XGBoost (ensemble of classification and regression trees (CARTs))	Adult Census Income, MNIST
[47]	2016	1	0	0	1	0	0	CNN, LSTM, NN	MNIST, Works of Shakespeare
[48]	2017	0	1	0	1	0	0	LSTM	Reddit comments
[49]	2018	0	1	0	1	0	0	CNN, GRU	CSI, FaceScrub, FourSquare, Labeled Faces in the Wild (LFW), PIPA Flickr images, Yelp-health, Yelp-author
[50]	2019	1	0	0	0	0	0	Logistic Regression, LSTM	Adult Census Income, Cornell movie dataset, Fashion-MNIST, Penn Treebank
[51]	2018	0	1	0	0	0	0	CNN, NN	CIFAR-100, Purchase100, Texas100

Table 2. (continued) Abbreviations:
Ref.: Reference; Alg.: Algorithm; S/P: Security/Privacy; Comm.: Communication Efficiency;
DP: Differential Privacy; MPC: Multi-Party Computation; HE: Homomorphic Encryption

Ref.	Year	Alg.	S./P.	Comm.	DP	MPC	HE	ML Models	Data
[52]	2018	1	0	1	0	0	0	CNN	CIFAR-10, FMNIST
[53]	2018	1	0	0	0	0	0	Linear Models	17 datasets from UCI
[54]	2018	0	1	0	1	0	0	CNN	OpenImages, PIPA Flickr images
[55]	2018	1	0	1	0	0	0	Logistic Regression, LSTM	FEMNIST, MNIST, Synthetic data, Twitter Sentiment140, Works of Shakespeare
[56]	2019	1	0	0	0	0	0	LMS Filter	MHEALTH
[57]	2019	0	0	1	0	0	0	CNN, Logistic Regression, LSTM	CIFAR-10, Fashion-MNIST, MNIST, Speech Commands
[58]	2018	0	0	0	0	0	0	CNN	BraTS (Brain Tumor Segmentation)
[59]	2017	1	0	1	0	0	0	SVM	Google Glass (GLEAM), UCI smartphone HAR, Vehicle Sensors Network
[60]	2018	1	0	0	0	0	0	NN, Logistic Regression (Multi-Class)	Real-Life heterogeneity HAR dataset
[61]	2019	1	0	1	0	0	0	Elastic Net Logistic Regression, Kernel-SVM, Random Forest	OpenML Bioresponse, OpenML fri_c4_500_100, OpenML Gina Agnostic
[62]	2017	0	0	1	0	0	0	K-Means, Distributed Power Iteration	CIFAR-10, MNIST
[63]	2019	0	0	1	0	0	0	-	-
[64]	2018	1	1	1	1	1	1	CNN, Decision Tree	MNIST, UCI Nursery dataset
[65]	2018	1	0	1	0	0	0	CNN	CIFAR-10, CIFAR-100
[66]	2018	1	0	0	1	0	0	GRU, NN	Keyboard metadata from BiAffect study about mood disturbance

Table 2. (continued) Abbreviations:
Ref.: Reference; Alg.: Algorithm; S/P: Security/Privacy; Comm.: Communication Efficiency;
DP: Differential Privacy; MPC: Multi-Party Computation; HE: Homomorphic Encryption

Ref.	Year	Alg.	S./P.	Comm.	DP	MPC	HE	ML Models	Data
[67]	2019	0	0	0	0	0	0	CNN, LSTM	MNIST, UCI Semeion Handwritten Digits, UCI Smartphone HAR, Works of Shakespeare
[68]	2019	1	0	1	0	0	0	CNN, K-Means, Linear Regression, Squared SVM	Fashion-MNIST, MNIST, energy consumption dataset, user knowledge modelling dataset
[69]	2018	0	1	0	0	0	0	CNN	AT&T Faces, MNIST
[70]	2019	1	0	1	0	0	0	CNN	CIFAR-10
[71]	2018	1	0	1	0	0	0	CNN	AudioSet, CIFAR-10, Google Speech Commands, Toxic comments
[72]	2019	1	0	0	0	0	0	-	-
[73]	2018	0	0	1	0	0	0	SVM	CIFAR-10
[74]	2019	1	1	1	1	1	1	-	-
[75]	2018	1	0	1	0	0	0	CNN	CIFAR-10, MNIST
[76]	2017	1	1	1	1	0	1	CNN, NN	MNIST, Street View House Numbers (SVHN)
[77]	2018	1	0	0	0	0	0	CNN	CIFAR-10, MNIST, Speech Commands
[78]	2018	0	1	1	1	0	0	CNN, PCA	CIFAR-10, MNIST
[79]	2018	1	0	1	0	0	0	CNN, NN	MNIST
[80]	2019	1	0	0	0	0	0	NN	MNIST

C SEARCH TERMS

Table 3. Exact search queries

Search Engine	Query	# Results
ACM Full-Text Collection	"query": { "federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated optimisation" OR "federated optimization" OR "federated SGD" }	12
Arxiv	all="federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated optimisation" OR "federated optimization" OR "federated SGD"	106
IEEE Xplore	"federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated optimisation" OR "federated optimization" OR "federated SGD"	46
PubMed	("machine learning"[All Fields] OR "deep learning"[All Fields] OR "artificial intelligence"[All Fields]) AND federated[Title/Abstract]	13
Web of Science	TOPIC: ("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated optimisation" OR "federated optimization" OR "federated SGD") OR TITLE: ("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated optimisation" OR "federated optimization" OR "federated SGD")	43