A Systematic Literature Review on Federated Machine Learning: From a Software Engineering Perspective

SIN KIT LO and QINGHUA LU, Data61, CSIRO and University of New South Wales, Australia
CHEN WANG, Data61, CSIRO, Australia
HYE-YOUNG PAIK, University of New South Wales, Australia
LIMING ZHU, Data61, CSIRO and University of New South Wales, Australia

Federated learning is an emerging machine learning paradigm where clients train models locally and formulate a global model based on the local model updates. To identify the state-of-the-art in federated learning and explore how to develop federated learning systems, we perform a systematic literature review from a software engineering perspective, based on 231 primary studies. Our data synthesis covers the lifecycle of federated learning system development that includes background understanding, requirement analysis, architecture design, implementation, and evaluation. We highlight and summarise the findings from the results and identify future trends to encourage researchers to advance their current work.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Multi-agent systems**;

Additional Key Words and Phrases: Federated learning, systematic literature review, software engineering, distributed learning, edge learning, privacy

ACM Reference format:
Sin Kit Lo, Qinghua Lu, Chen Wang, Hye-Young Paik, and Liming Zhu. 2021. A Systematic Literature Review on Federated Machine Learning: From a Software Engineering Perspective. *ACM Comput. Surv.* 54, 5, Article 95 (May 2021), 39 pages.
https://doi.org/10.1145/3450288

1 INTRODUCTION

Machine learning is adopted broadly in many areas, and data plays a critical role in machine learning systems due to its impact on the model performance. Although the widely deployed remote devices (e.g., mobile/IoT devices) generate massive amounts of data, data hungriness is still a challenge because of the increasing concern in data privacy (e.g., **General Data Protection Regulation (GDPR)** [3]).

To effectively address this challenge, federated learning was proposed by Google in 2016 [113]. In federated learning, client devices perform model training locally and generate a global model

Authors' addresses: S. K. Lo, Q. Lu, and L. Zhu, Data61, CSIRO and University of New South Wales, Australia; emails: {Kit.Lo, qinghua.lu, Liming.Zhu}@data61.csiro.au; C. Wang, Data61, CSIRO, Australia; email: Chen.Wang@data61.csiro.au; H.-Y. Paik, University of New South Wales, Australia; email: h.paik@unsw.edu.au.

0360-0300/2021/05-ART95 $15.00
https://doi.org/10.1145/3450288

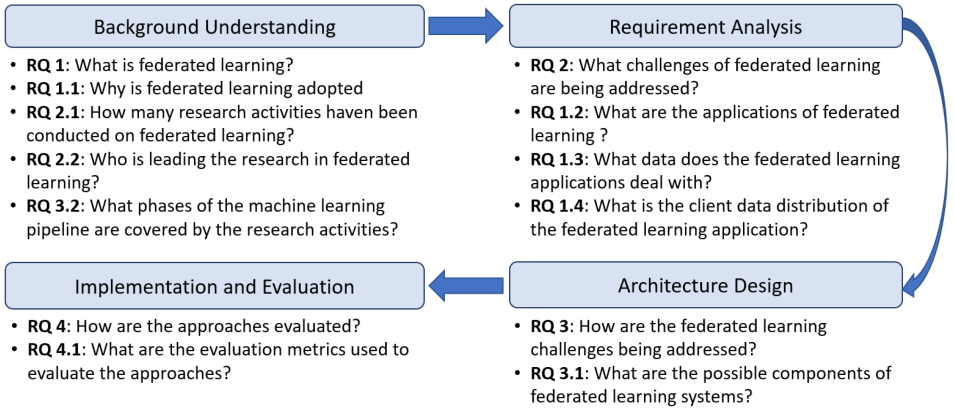

Fig. 1. Research questions mapping with Software Development Life Cycle (SDLC).

collaboratively. The data is stored locally and is never transferred to the central server or other clients [39, 73]. Instead, only model updates are communicated for formulating the global model.

The growing interest in federated learning has increased the number of research projects and publications. Although many surveys are conducted on this topic [73, 93, 95], there is still no systematic literature review on federated learning. It motivates us to perform a systematic literature review on federated learning to understand the state-of-the-art. Furthermore, client devices in federated learning systems form a large-scale distributed system. It calls for software engineering considerations apart from the core machine learning knowledge [163]. Thus, we explore how to develop federated learning systems by conducting a systematic literature review to provide a holistic and comprehensive view of the state-of-the-art federated learning research, especially from a software engineering perspective.

We perform a systematic literature review following Kitchenham's standard guideline [81]. The objectives are to: (1) provide an overview of the research activities and diverse research topics in federated learning system development; (2) help practitioners to understand the challenges and approaches to develop a federated learning system. The contributions of this article are as follows:

- We present a comprehensive qualitative and quantitative synthesis reflecting the state-of-the-art in federated learning with data extracted from 231 primary studies. Our data synthesis investigates different stages of federated learning system development.
- We provide all the empirical findings and identify the future trends in federated learning research.

The remainder of the article is organised as follows: Section 2 introduces the methodology. Section 3 presents the results and highlights the findings. Section 4 identifies future trends, followed by threats to validity in Section 5. Section 6 discusses related work. Section 7 concludes the article.

2 METHODOLOGY

Based on Kitchenham's guideline [81], we developed the following protocol.

2.1 Research Questions

To provide a systematic literature review from the software engineering perspective, we view federated learning as a software system and study federated learning systems from the software development standpoint. As shown in Figure 1, we adopt the software development practices of

machine learning systems in Reference [163] to describe the **software development lifecycle (SDLC)** for federated learning. In this section, we explain how each **research question (RQ)** is derived.

2.1.1 Background Understanding. To develop a federated learning system, we need to first know what federated learning is and when to adopt federated learning instead of centralised learning. Thus, we derive **RQ 1 (What is federated learning?)** and **RQ 1.1 (Why is federated learning adopted?)**. After learning the background and adoption objectives of federated learning, we identify the research efforts and the experts in this field through **RQ 2.1 (How many research activities have been conducted on federated learning?)** and **RQ 2.2 (Who is leading the research in federated learning?)**. Last, we derive **RQ 3.2 (What are the phases of the machine learning pipeline covered by the research activities?)** to examine the focused machine learning pipeline stages in the existing studies and understand the maturity of the area.

2.1.2 Requirement Analysis. After the background understanding stage, the requirements of federated learning systems are analysed. We focus on the non-functional requirements, since functional requirements are application-specific. We derive **RQ 2 (What challenges of federated learning are being addressed?)** to identify the architectural drivers (i.e., non-functional requirements) of federated learning systems. **RQ 1.2 (What are the applications of federated learning?)**, **RQ 1.3 (What data does the federated learning applications deal with?)**, and **RQ 1.4 (What is the client data distribution of the federated learning applications?)** are designed to help researchers and practitioners assess the suitability of federated learning for their systems, which is within the scope of the requirement analysis.

2.1.3 Architecture Design. After the requirement analysis, researchers and practitioners need to understand how to design the architecture. For this, we consider the approaches against each requirement. Hence, we derive **RQ 3 (How are the federated learning challenges being addressed?)** and **RQ 3.1 (What are the possible components of federated learning systems?)**. These 2 RQs aim (1) to identify the possible approaches that address the challenges during the federated learning system development, and (2) to extract the software components for federated learning architecture design to fulfill the non-functional requirements (i.e., challenges).

2.1.4 Implementation and Evaluation. After the architecture design stage, once the system is implemented, the federated learning systems including the built models need to be evaluated. Thus, we derive **RQ 4 (How are the approaches evaluated?)** and **RQ 4.1 (What are the evaluation metrics used to evaluate the approaches?)** to identify the methods and metrics for the evaluation of federated learning systems.

2.2 Sources Selection and Strategy

We searched through the following search engines and databases: (1) *ACM Digital Library*, (2) *IEEE Xplorer*, (3) *ScienceDirect*, (4) *Springer Link*, (5) *ArXiv*, and (6) *Google scholar*. The search time frame is set between 2016.01.01 and 2020.01.31. We screened and selected the papers from the initial search according to the preset inclusion and exclusion criteria elaborated in Section 2.2.2.

We then conducted forward and backward snowballing processes to search for any related papers that were left out from the initial search. The paper selection process consists of two phases: (1) The papers were first selected by two researchers through title and abstract screening independently, based on the inclusion and exclusion criteria. Then, the two researchers cross-checked the results and resolved any disagreement on the decisions. (2) The papers selected in the first phase were then assessed through full-text screening. The two researchers again cross-checked the selection results and resolved any disagreement on the selection decisions. Should any disagreement

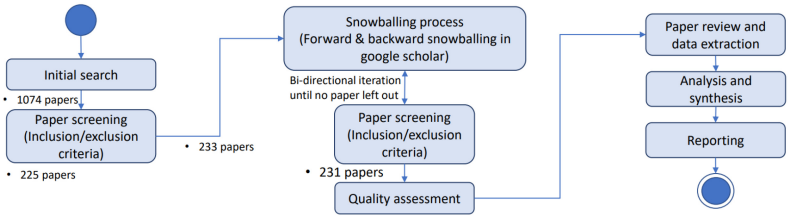

Fig. 2. Paper search and selection process map.

Table 1. Number of Selected Publications per Source

Sources	ACM	IEEE	Springer	ScienceDirect	ArXiv	Google Scholar	Total
Paper count	22	74	17	4	106	8	231

Table 2. Key and Supplementary Search Terms

Key Term	Supplementary Terms
Federated Learning	Federated Machine Learning, Federated ML, Federated Artificial Intelligence, Federated AI, Federated Intelligence, Federated Training

Table 3. Search Strings and Quantity of *ACM Digital Library*

Search string	All: "federated learning"] OR [All: ,] OR [All: "federated machine learning"] OR [All: "federated ml"] OR [All: ,] OR [All: "federated intelligence"] OR [All: "federated training"] OR [All: ,] OR [All: "federated artificial intelligence"] OR [All: ,] OR [All: "federated ai"] AND [Publication Date: (01/01/2016 TO 01/31/2020)]
Result quantity	76
Selected papers	**22**

have occurred in either the first or second phase, a third researcher (meta-reviewer) was consulted to finalise the decision. Figure 2 shows the paper search and selection process.

The initial search found 1,074 papers, with 76 from *ACM Digital Library*, 320 from *IEEE Xplorer*, 5 from *ScienceDirect*, 85 from *Springer Link*, 256 from *ArXiv*, and 332 from *Google scholar*. After the paper screening, exclusion, and duplicates removal, we ended up with 225 papers. From there, we conducted the snowballing process and found 6 more papers. The final paper number for this review is 231. The number of papers per source are presented in Table 1.

2.2.1 Search String Definition. We used "Federated Learning" as the key term and included synonyms and abbreviations as supplementary terms to increase the search results. We designed the search strings for each primary source to check the title, abstract, and keywords. After completing the first draft of search strings, we examined the results of each search string against each database to check the effectiveness of the search strings. The finalised search terms are shown in Table 2. The search strings and the respective paper quantities of the initial search for each primary source are shown in Tables 3, 4, 5, 6, 8, and 7.

Table 4. Search Strings and Quantity of *IEEE Xplorer*

Search string	(“Document Title”:“federated learning” OR “federated training” OR “federated intelligence” OR “federated machine learning” OR “federated ML” OR “federated artificial intelligence” OR “federated AI”) OR (“Author Keywords”:“federated learning” OR “federated training” OR “federated intelligence” OR “federated machine learning” “federated ML” OR “federated artificial intelligence” OR “federated AI”)
Result quantity	320
Selected papers	**71**

Table 5. Search Strings and Quantity of *ScienceDirect*

Search string	“federated learning” OR “federated intelligence” OR “federated training” OR “federated machine learning” OR “federated ML” OR “federated artificial intelligence” OR “federated AI”
Result quantity	5
Selected papers	**4**

Table 6. Search Strings and Quantity of *Springer Link*

Search string	“federated learning” OR “federated intelligence” OR “federated training” OR “federated machine learning” OR “federated ML” OR “federated artificial intelligence” OR “federated AI”
Result quantity	85
Selected papers	**17**

2.2.2 Inclusion and Exclusion Criteria. The inclusion and exclusion criteria are formulated to effectively select relevant papers. After completing the first draft of the criteria, we conducted a pilot study on 20 randomly selected papers. Then, the two independent researchers cross-validated the papers selected by the other researcher and refined the criteria.

The finalised inclusion criteria are as follows:

- Both long and short papers that have elaborated on the component interactions of the federated learning system: We specifically focus on the research works that provide comprehensive explanations on the federated learning components functionalities and their mutual interactions.
- Survey, review, and SLR papers: We included all the surveys and review papers to identify the open problems and future research trends in an objective manner. However, we excluded them from the stages for research questions analyses.
- We included ArXiv and Google scholar’s papers cited by the peer-reviewed papers published in the primary sources.

The finalised exclusion criteria are as follows:

- Papers that elaborate only low-level communication algorithms: The low-level communication algorithms or protocols between hardware devices are not the focus of this work.

- Papers that focus only on pure gradient optimisation. We excluded the papers that purely focus on the gradient and algorithm optimisation research. Our work focuses on the multi-tier processes and interactions of the federated learning software components.
- Papers that are not in English.
- Conference version of a study that has an extended journal version.
- PhD dissertations, tutorials, editorials and magazines.

2.2.3 Quality Assessment. A quality assessment scheme was developed to evaluate the quality of the papers. There are four quality criteria (QC) used to rate the papers. We used numerical scores ranging from 1.00 (lowest) to 5.00 (highest) to rate each paper. The average scores are calculated for each QC and the total score of all four QCs are obtained. We included papers that score greater than 1.00 to avoid missing out on studies that are insightful while maintaining the quality of the search pool. The QCs are as follows:

- QC1: The citation rate. We identified this by checking the number of citations received by each paper according to *Google scholar*.
- QC2: The methodology contribution. We identified the methodology contribution of the paper by asking 2 questions: (1) Is this paper highly relevant to the research? (2) Can we find a clear methodology that addresses its main research questions and goals?
- QC3: The sufficient presentation of the findings. Are there any solid findings/results and clear-cut outcomes? Each paper is evaluated based on the availability of results and the quality of findings.
- QC4: The future work discussions. We assessed each paper based on the availability of discussions on future work.

2.2.4 Data Extraction and Synthesis. We downloaded all the selected papers and recorded all the essential information in a data extraction sheet,[1] including the title, source, year, paper type, venue, authors, affiliation, the number of citations of the paper, the score from all QCs, the answers for each RQ, and the research classification. The following steps were followed to prevent any data extraction bias:

- The two independent researchers conducted the data extraction of all the papers and cross checked the classification and discussed any dispute or inconsistencies in the extracted data.
- For any unresolved dispute on the extracted data, the first two authors tried to reach an agreement. When an agreement was not met, the meta reviewer reviewed the paper and finalised the decision together.
- All the data was recorded in the Google sheet for analysis and synthesis processes.

3 RESULTS

In this section, the extracted results of each research question are summarised and analysed.

3.1 RQ 1: What is Federated Learning?

The first research question (RQ 1) is "What is federated learning?" To answer RQ 1, the definition of federated learning reported by each study is recorded. This question helps the audience to understand: (1) what federated learning is, and (2) the perceptions of researchers on federated learning. Figure 3 is a word cloud that shows the frequency of the words that appear in the original definition of federated learning in each study. The most frequently appeared words include:

[1]Data extraction sheet, https://drive.google.com/file/d/10yYG8W1FW0qVQOru_kMyS86owuKnZPnz/view?usp=sharing.

Fig. 3. RQ 1: What is federated learning?

Table 7. Search Strings and Quantity of *ArXiv*

Search string	order: -announced_date_first; size: 200; date_range: from 2016-01-01 to 2020-01-31 include_cross _list:True; terms: AND title="federated learning" OR "federated intelligence" OR "federated training" OR "federated machine learning" OR "federated ML" OR "federated artificial intelligence" OR "federated AI"; OR abstract="federated learning" OR "federated intelligence" OR "federated training" OR "federated machine learning" OR "federated ML" OR "federated artificial intelligence" OR "federated AI"
Result quantity	256
Remark	Search title and abstract only (*ArXiv* does not provide keyword search option)
Selected papers	**103**

Table 8. Search Strings and Quantity of *Google Scholar*

Search string	"federated learning" OR "federated intelligence" OR "federated training" OR "federated machine learning" OR "federated ML" OR "federated artificial intelligence" OR "federated AI"
Result quantity	332
Remark	Search title only (*Google scholar* does not provide abstract and keyword search option)
Selected papers	**8**

distribute, local, device, share, client, update, privacy, aggregate, edge, and so on. To answer RQ1 more accurately, we use these five categories to classify the definitions of federation learning: (1) training settings, (2) data distribution, (3) orchestration, (4) client types, and (5) data partitioning, as understood by the researchers, as shown in Table 9.

First, in the training settings, building a decentralised or distributed learning process over multiple clients is what most researchers conceived as federated learning. This can be observed as the most frequently mentioned keyword in RQ1 is "training a model on multiple clients." Other frequently mentioned keywords that describe the training settings are "distributed," "collaborative," and "multiple parties/clients." "Only sending the model updates to the central server" and "producing a global model on the central server" are the other two characteristics that describe how a federated learning system performs model training in a distributed manner. This also shows how

Table 9. Characteristics of Federated Learning

Category	Characteristic	Paper count
Training settings (82%)	Training a model on multiple clients	200
	Only sending model updates to the central server	133
	Producing a global model on the central server	63
Data distribution (11%)	Decentralised data storage	40
	Data generated locally	17
Orchestration (3%)	Training organised by a central server	13
Client types (3%)	Cross-device	11
	Cross-silo	1
	Both	3
Data partitioning (<1%)	Horizontal federated learning	1
	Vertical federated learning	1

researchers differentiate federated learning from conventional machine learning and distributed machine learning.

Second, federated learning can be explained in terms of the data distributions. The keywords mentioned in the studies are "data generated locally" and "data stored decentralised." Data is collected and stored by client devices in different geographical locations. Hence, it exhibits non-IID (non-Identically and Independently Distributed) and unbalanced data properties [73, 113]. Furthermore, the data is decentralised and is not shared with other clients to preserve data privacy. We will discuss more on the client data distribution in Section 3.4.

Third, researchers observe federated learning from the training process orchestration standpoint. In conventional federated learning, a central server orchestrates the training processes. The tasks consist of initialisation of a global model, distribution of the global models to participating client devices, collection of trained local models, and the aggregation of the collected local models to update the global model. Intuitively, researchers consider the usage of a single central server as a possible single-point-of-failure [110, 143]. Hence, decentralised approaches for the exchange of model updates are studied and the adoption of blockchains for decentralised data governance is introduced [110, 143, 210].

Fourth, we observe two types of federated learning in terms of client types that are cross-device and cross-silo. Cross-device federated learning deals with a massive number of smart devices, creating a large-scale distributed network to collaboratively train a model for the same applications [73]. Some examples of the applications are mobile device keyboard word suggestions and human activity recognition. The setting are extended to cross-silo applications where data sharing between organisations is prohibited. For instance, data of a hospital is prohibited from exposure to other hospitals due to data security regulations. To enable machine learning under this environment, cross-silo federated learning conducts local model training using the data in each hospital (silo) [73, 160, 211].

Last, we found three data partitioning variations: horizontal, vertical, and federated transfer learning. Horizontal federated learning, or sample-based federated learning, is used when the datasets share the same feature space but different sample ID space [103, 183]. Inversely, vertical federated learning, also known as feature-based federated learning, is applied to the cases where two or more datasets share the same sample ID space but different feature space. Federated transfer learning considers the data partitioning where two datasets only overlap partially in the sample space or the feature space. It aims to develop models that are applicable to both datasets [183].

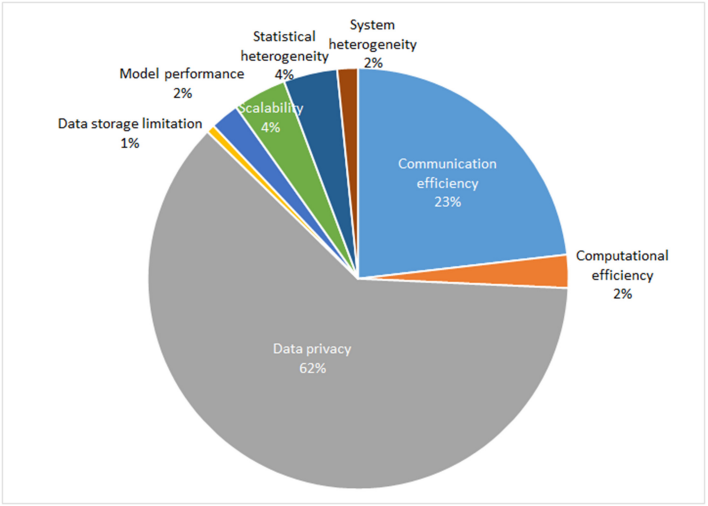

Fig. 4. RQ 1.1: Why federated learning is adopted?

We summarised the findings as below. We connected each keyword and grouped them under the same definition. Finally, we arranged the definitions according to the frequency of the words that appeared.

Findings of RQ 1: What is federated learning (FL)?

Federated learning is a type of distributed machine learning to preserve data privacy. Federated learning systems rely on a central server to coordinate the model training process on multiple, distributed client devices where the data are stored. The model training is performed locally on the client devices, without moving the data out of the client devices. Federated learning can also be performed in a decentralised manner.

Variations in federated learning: (1) centralised/decentralised federated learning, (2) cross-silo/device federated learning, (3) horizontal/vertical/transfer federated learning.

With regard to the software development lifecycle, this question contributes to the *background understanding* phase where we provide the definition of the fundamental settings and different variations of federated learning as reported by researchers and practitioners.

3.2 RQ 1.1: Why is Federated Learning Adopted?

The motivation of RQ 1.1 is to understand the advantages of federated learning. We classify the answers based on the non-functional requirements of federated learning adoption (illustrated in Figure 4). Data privacy and communication efficiency are the two main motivations to adopt federated learning. Data privacy is preserved in federated learning as no raw local data moves out of the device [20, 135]. Also, federated learning achieves higher communication efficiency by exchanging only model parameters or gradients [73, 113, 183]. High data privacy and communication efficiency also promote scalability. Hence, more clients are motivated to join the training process [73, 113, 183].

Statistical heterogeneity is defined as the data distribution with data volume and class distribution variance among devices (i.e., Non-IID). Essentially, the data are massively distributed across client devices, each only contains small amount of data [119, 138, 198], with unbalanced data classes [119] that are not representative of the overall data distribution [64]. When local

models are trained independently on these devices, these models tend to be over-fitted to their local data [97, 138, 198]. Hence, federated learning is adopted to collaboratively trains the local models to form a generalised global model. System heterogeneity is defined as the property of devices having heterogeneous resources (e.g., computation, communication, storage, and energy). Federated learning can tackle this issue by enabling local model training and only communicates the model updates, which reduce the bandwidth footprint and energy consumption [7, 36, 97, 119, 136, 154, 166].

Another motivation is high computation efficiency. With a large number of participating clients and the increasing computation capability of clients, federated learning can have high model performance [12, 38, 113, 150, 205] and computation efficiency [11, 62, 198]. Data storage efficiency is ensured by independent on-client training using locally generated data [22, 113, 146, 194, 208].

Findings of RQ 1.1: Why is federated learning adopted?

Motivation for adoption: Data privacy and communication efficiency are the two main motivations. Only a small number of studies adopt federated learning because of model performance. With a large number of participating clients, federated learning is expected to achieve high model performance. However, the approach is still immature when dealing with non-IID and unbalanced data distribution.

With regard to the software development lifecycle, this question contributes to the *background understanding* phase where we identify the objectives of federated learning adoption.

3.3 RQ 1.2 What are the Applications of Federated Learning? RQ 1.3: What Data Does the Federated Learning Applications Deal with?

We study the federated learning applications and the types of data used in those applications through RQ 1.2 and RQ 1.3. The data types and applications are listed in Table 10. The most widely used data types are image data, structured data, and text data, while the most popular application is image classification. In fact, MNIST[2] is the most frequently used dataset. More studies are needed to deal with IoT time-series data. Both graph data and sequential data are not popularly used in federated learning due to their data characteristics. We observed that the federated learning is widely adopted in applications that infer personal data, such as images, personal medical or financial data, and text recorded by personal mobile devices.

Findings of RQ 1.2: What are the applications of federated learning?
RQ 1.3: What data does the federated learning applications deal with?

Applications and data: Federated learning is widely adopted in applications that deal with image data, structured data, and text data. Both graph data and sequential data are not popularly used due to their data characteristics (e.g., non-linear data structure). Also, there is only a few production-level applications. Most applications are still proof-of-concept prototypes or simulations.

With regard to the software development lifecycle, this question contributes to the *requirement analysis* phase where we identify the different applications and data types that have applied federated learning as a reference for researchers and practitioners.

[2]The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/.

Table 10. Data Types and Applications Distribution

Data Types (RQ 1.3)	Applications (RQ 1.2)	Count
Graph data (5%)	Generalized Pareto Distribution parameter estimation	1
	Incumbent signal detection model	1
	Linear model fitting	1
	Network pattern recognition	8
	Computation resource management	1
	Waveform classification	1
Image data (49%)	Autonomous driving	5
	Healthcare (Bladder contouring, whole-brain segmentation)	2
	Clothes type recognition	14
	Facial recognition	2
	Handwritten character/digit recognition	109
	Human action prediction	2
	Image processing (classification/defect detection)	4
	Location prediction	1
	Phenotyping system	1
	Content recommendation	2
Sequential data (4%)	Game AI model	2
	Network pattern recognition	1
	Content recommendation	1
	Robot system navigation	1
	Search rank system	6
	Stackelberg competition model	2
Structured data (21%)	Air quality prediction	2
	Healthcare	25
	Credit card fraud detection	3
	Bankruptcy prediction	5
	Content recommendation (e-commerce)	1
	Energy consumption prediction	1
	Economic prediction (financial/house price/income/loan/market)	11
	Human action prediction	4
	Multi-site semiconductor data fusion	1
	Particle collision detection	1
	Industrial production recommendation	1
	Publication dataset binary classification	1
	Quality of Experience (QoE) prediction	1
	Search rank system	1
	Sentiment analysis	1
	System anomaly detection	1
	Song publishment year prediction	1

(Continued)

Table 10. Continued

Data Types (RQ 1.3)	Applications (RQ 1.2)	Count
	Customer satisfaction prediction	1
	Keyboard suggestion (search/word/emoji)	21
	Movie rating prediction	4
	Out-of-Vocabulary word learning	1
	Suicidal ideation detection	1
	Product review prediction	1
Text data (14%)	Resource management model	1
	Sentiment analysis	5
	Spam detection	2
	Speech recognition	1
	Text-to-Action decision model	1
	Content recommendation	1
	Wine quality prediction	1
	Air quality prediction	1
	Automobile MPG prediction	1
	Healthcare (gestational weight gain / heart rate prediction)	2
	Energy prediction (consumption/demand/generation)	3
Time-series data	Human action prediction	8
(7%)	Location prediction	1
	Network anomaly detection	1
	Resource management model	2
	Superconductors critical temperature prediction	1
	Vehicle type prediction	1

Table 11. Client Data Distribution Types of Federated Learning Applications

Data distribution types	Non-IID	IID	Both	N/A
Percentages	24%	23%	13%	40%

3.4 RQ 1.4: What is the Client Data Distribution of the Federated Learning Applications?

Table 11 shows the client data distribution types found in the studies. 24% of the studies have adopted non-IID data or have addressed the non-IID issue in their work. 23% of the studies have adopted IID data. 13% of the studies have compared the two types of client data distributions (Both), whereas 40% of the studies have not specified which data distribution they have adopted (N/A). These studies ignored the effect of data distribution on the federated model performance. In the simulated non-IID data distribution settings, researchers mainly split the dataset by class and store each class into different client devices (e.g., References [31, 66, 67, 153, 202]), or by sorting the data accordingly before distributing them to each client device [114]. Furthermore, the data volume is uneven for each client device [67, 80]. For use case evaluations, the non-IID data are generated or collected by local devices, such as References [42, 45, 57, 84, 134]. For IID data distribution settings, the data are randomised and distributed evenly to each client device (e.g., References [13, 72, 114, 153, 202]).

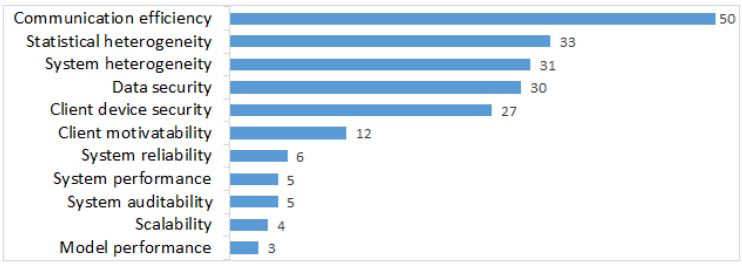

Fig. 5. Challenges of federated learning.

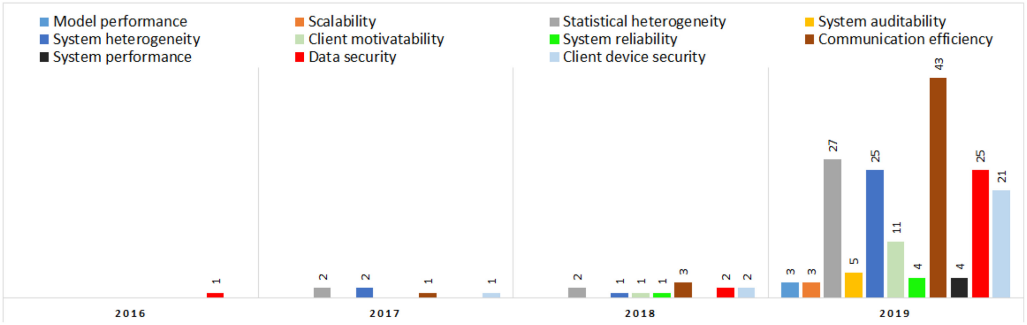

Fig. 6. Research Area Trend.

Findings of RQ 1.4:
What is the client data distribution of the federated learning applications?
Client data distribution: The client data distribution influences the federated learning model performance. Model aggregation should consider that the distribution of the dataset on each client is different. Many studies are conducted on Non-IID issues, particularly on the FedAvg algorithm extensions for model aggregation.
With regard to the software development lifecycle, this question contributes to the *requirement analysis* phase where we identify the characteristics of the different types of data distribution that affect the federated learning model performance.

3.5 RQ 2: What Challenges of Federated Learning are Being Addressed? RQ 2.1: How Many Research Activities have been Conducted on Federated Learning?

The motivation of RQ 2 is to identify the challenges of federated learning that are addressed by the studies. As shown in Figure 5, we group the answers into categories based on ISO/IEC 25010 System and Software Quality model [2] and ISO/IEC 25012 Data Quality model [1]. We can observe that the communication efficiency of federated learning received the most attentions from researchers, followed by statistical heterogeneity and system heterogeneity.

To explore the research interests evolved from 2016 to 2019, we cross-examined the results for RQ 2 and RQ 2.1 as illustrated in Figure 6. Note that we included the results from 2016 to 2019 as we only searched studies up to 31.01.2020 and the trend of one month does not represent the trend of the entire year. We can see that the number of studies on communication efficiency, statistical heterogeneity, system heterogeneity, data security, and client device security surged drastically in 2019 compared to the years before.

Although transferring model updates instead of raw data can reduce communication costs, federated learning systems still perform multiple update iterations to reach convergence. Therefore, ways to reduce communication rounds are studied (e.g., References [41, 114, 148]). Furthermore, cross-device federated learning needs to accommodate a large number of client devices. Hence, some clients may drop out due to bandwidth limitation [40, 175]. These dropouts could negatively impact the outcome of federated learning systems in two ways: (i) reduce the amount of data available for training [40], (ii) increase the overall training time [111, 147]. The mechanism in Reference [104] addresses the dropout problem by abandoning the global model aggregated from a low number of local models.

Federated learning is effective in dealing with statistical and system heterogeneity issues through the aggregation of local models trained locally on client devices [36, 97]. However, the approach is still immature as open questions exist in handling the non-IID while maintaining the model performance [36, 66, 158, 161, 190].

The interests in data security (e.g., References [18, 49]) and client device security (e.g., References [35, 48, 171]) are also high. The data security in federated learning is the degree to which a system ensures data are accessible only by an authorised party [1]. While federated learning systems restrict raw data from leaving local devices, it is still possible to extract private information through the back-tracing of gradients. The studies under this category mostly express their concerns about the possibility of information leakage from the local model gradient updates. Client device security can be expressed as the degree of security against dishonest and malicious client devices [73]. The existence of malicious devices in the training process could poison the overall model performance by disrupting the training process or providing false updates to the central server. Furthermore, the intentional or unintentional misbehavior of client devices could reduce system reliability.

Client motivatability is discussed as an aspect to be explored (e.g., References [79, 80, 196]), since model performance relies greatly on the number of participating clients. More participating clients means more data and computation resources are contributed to the model training process.

System reliability concerns are mostly on the adversarial or byzantine attacks that target the central server, hence exposes the single-point-of-failure (e.g., References [63, 87, 110]). The system performance of federated learning systems is mentioned in some studies (e.g., References [34, 106, 174]), which includes the considerations on computation efficiency, energy usage efficiency, and storage efficiency. In resource-restricted environments, the efficient resource management of federated learning systems is more crucial to system performance.

To improve system auditability, auditing mechanisms (e.g., References [8, 72, 170]) are used to track the client devices' behavior, local model performance, and system runtime performance. Scalability is also mentioned as a limitation in federated learning (e.g., References [135, 142, 143, 174]) and, last, the model performance limitation is investigated (e.g., References [64, 69, 123]). The model performance of federated learning systems is highly dependent on the number of participants and the data volume of each participating client in the training process. Moreover, there is model performance limitation due to the non-IID data.

Findings of RQ 2: What challenges of federated learning are being addressed?
Motivation vs. challenges: Most of the known motivations of federated learning also appear to be the most studied federated learning limitations, including communication efficiency, system and statistical heterogeneity, model performance, and scalability. This reflects that federated learning is still under-explored.
With regard to the software development lifecycle, this question contributes to the *requirement analysis* phase where we identify the various requirements of a federated learning system to be considered during the development.

Table 12. The Research Classification of the Selected Paper

Research Classification	Evaluation research	Philosophical paper	Proposal of solution	Validation research
Paper count	13	12	25	183

To answer RQ 2.1, we classify the papers according to the research classification criteria proposed by Wieringa [172], which includes: (1) evaluation research, (2) philosophical papers, (3) proposal of solution, and (4) validation research. We use this classification to distinguish the focus of each research activity. *Evaluation research* is the investigation of a problem in software engineering practice. In general, the research results in new knowledge of causal relationships among phenomena, or in new knowledge of logical relationships among propositions. The causal properties are studied through case or field studies, field experiments and surveys. *Philosophical papers* propose a new way of looking at things, for instance, a new conceptual framework. *Proposal of solution* papers propose solution techniques and argue for its relevance, but without a full-blown validation. Last, *validation research* investigates the properties of a proposed solution that has not yet been implemented in practice. The investigation uses a thorough, methodologically sound research setup (e.g., experiments, simulations).

The research classification results are presented in Table 12. The most common type of research is validation research, while the other types of are far less frequent. In particular, there are few philosophical papers that propose a new conceptual framework for federated learning.

Findings of RQ 2.1:
How many research activities have been conducted on federated learning?
Research activities: The number of studies that explored communication efficiency, statistical and system heterogeneity, data security, and client device security surged drastically in 2019 compared to the years before. The most conducted research activities are validation research, followed by proposal of solution, evaluation research, and philosophical papers.
With regard to the software development lifecycle, this question contributes to the *background understanding* phase where we identify the types of research activities on federated learning.

3.6 RQ 2.2: Who is Leading the Research in Federated Learning?

The motivation of RQ 2.2 is to understand the research impact in the federated learning community. We also intend to help researchers identify the state-of-the-art research in federated learning by selecting the top affiliations. As shown in Table 13, we listed the top 10 affiliations by the number of papers published and the number of citations. Google, IBM, CMU, and WeBank appear in both the top-10 lists. From this table, we can identify the research institutions that made the most effort on federated learning and those that made the most impact on the research domain in terms of citations.

Findings of RQ 2.2: Who is leading the research in federated learning?
Affiliations: Google, IBM, CMU, and WeBank appear in the top 10 affiliations list both by the number of papers and by the number of citations, which reflect that they made the most efforts on federated learning and also the most impact on the research domain.
With regard to the software development lifecycle, this question contributes to the *background understanding* phase where we provide the list of leading affiliations to help researchers and practitioners identify the state-of-the-art of federated learning.

Table 13. Research Impact Analysis

Top 10 affiliations by number of papers			Top 10 affiliations by number of citations		
Rank	**Affiliations**	**Paper count**	**Rank**	**Affiliations**	**No. of citations**
1	Google	21	1	Google	2,269
2	IBM	11	2	Stanford University	217
3	WeBank	8	3	ETH Zurich	130
3	Nanyang Technological University	8	4	IBM	122
5	Tsinghua University	6	5	Cornell University	101
5	Carnegie Mellon University	6	6	Carnegie Mellon University	98
7	Beijing University of Posts and Telecommunications	5	7	ARM	82
7	Kyung Hee University	5	8	Tianjin University	76
9	Chinese Academy of Sciences	4	9	University of Oulu	73
9	Imperial College London	4	10	WeBank	66

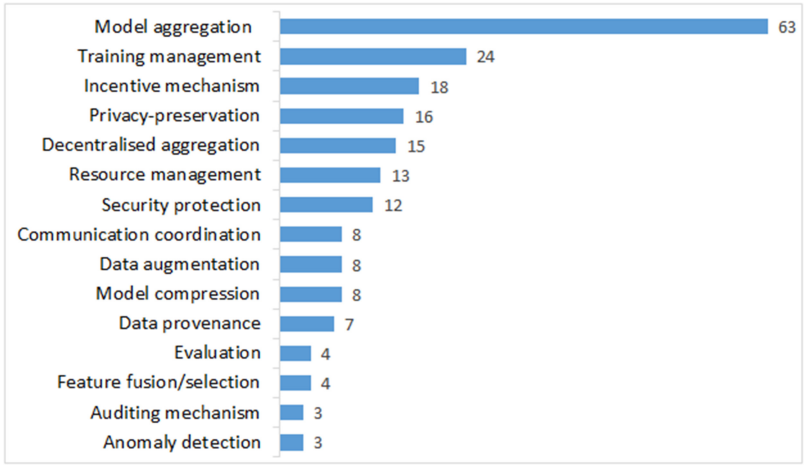

Fig. 7. RQ 3: How are the federated learning challenges being addressed?

3.7 RQ 3: How are the Federated Learning Challenges Being Addressed?

After looking at the challenges, we studied the approaches proposed by the researchers to address these challenges. Figure 7 shows the existing approaches: model aggregation (63), training management (24), incentive mechanisms (18), privacy-preserving mechanisms (16), decentralised aggregation (15), security management (12), resource management (13), communication coordination (8), data augmentation (7), data provenance (7), model compression (8), feature fusion/selection (4), auditing mechanisms (4), evaluation (4), and anomaly detection (3). We also mapped these approaches to the challenges of federated learning in Table 14. Notice that we did not include every challenge mentioned in the collected studies but only those that have a proposed solution.

As shown in Figure 7, model aggregation mechanisms are the most proposed solution by the studies. From Table 14, we can see that model aggregation mechanisms are applied to address communication efficiency, statistical heterogeneity, system heterogeneity, client device security, data security, system performance, scalability, model performance, system auditability, and system reliability issues.

Table 14. What Challenges of Federated Learning are Being Addressed (RQ 2) vs. How are the Federated Learning Challenges Being Addressed (RQ 3)

Challenges vs Approaches	Model performance	Scalability	Statistical heterogeneity	System auditability	System heterogeneity	Client motivatability	System reliability	Communication efficiency	System performance	Data security	Client device security
Model aggregation	[69]	[174],[135]	[160],[190],[37], [116],[174],[29], [101],[113],[153], [127],[96],[17], [77]	[8]	[190],[41],[116], [29],[194],[96], [19],[173],[122], [17]	-	[110]	[206],[32],[114],[131],[148], [207],[41],[103],[181],[180], [174],[29],[113],[68],[153], [20],[135],[72],[52],[19], [17],[149],[83]	[174],[29], [72]	[132],[19], [50]	[76],[129], [60],[30]
Training management	-	-	[152],[66],[23], [141],[39],[82], [89],[71],[51], [54],[36],[149], [53],[25]	-	[28],[39],[89], [53]	[47]	-	[188],[177],[197], [6],	-	-	[152]
Incentive mechanisms	-	-	[196]	-	[46],[140], [78]	[196],[80],[79], [171],[14],[75], [74],[46],[112], [78],[165]	-	[44],[126], [125]	-	-	-
Privacy preservation	-	-	-	-	-	-	-	-	-	[106],[178],[11], [109],[152],[115], [139],[49],[91], [156],[16],[169]	[158],[70], [197],[35]
Decentralised aggregation	-	[143]	-	-	[59]	-	[79],[143],[86], [87],[63]	[59]	[106]	[184]	[144],[138], [184],[85], [59]
Resource management	-	-	-	-	[168],[121], [9],[179], [193],[187], [97],[7], [137],[195], [92]	-	-	[141],[4]	-	-	-
Security protection	-	-	-	-	-	-	-	-	-	[198],[55],[56], [175],[111],[104], [58],[24],[33]	[175],[100], [102]
Communication coordination	-	-	-	-	-	-	-	[182],[65],[5], [27],[162],[147], [6]	[34]	-	-
Data augmentation	[123]	-	[42],[68]	-	-	-	-	[202]	-	[155], [128]	[124],[201]
Model compression	-	[142]	-	[18]	-	-	-	[167],[38], [142],[22], [82],[90]	-	-	-
Data provenance	-	-	-	-	-	-	-	-	-	[105],[110],[191]	[171],[107], [208],[203]
Evaluation	-	-	[21],[61]	[170]	[21]	-	-	-	-	-	-
Feature fusion/selection	[64]	-	[161]	[164]	-	-	-	[189]	-	-	-
Auditing mechanisms	-	-	-	[14],[10]	-	-	-	-	-	-	[14]
Anomaly detection	-	-	-	-	-	-	-	-	-	-	[130], [94], [48]

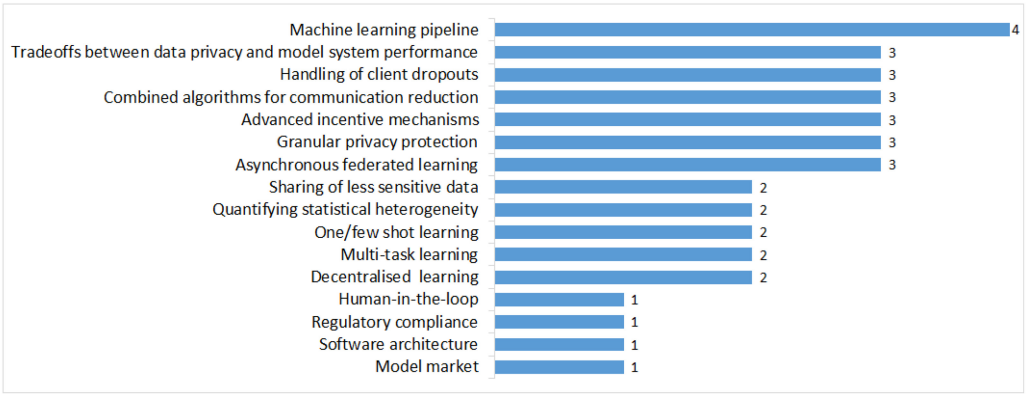

Fig. 8. Open problems and future research trends highlighted by existing reviews/surveys.

Researchers have proposed various kinds of aggregation methods, including selective aggregation [76, 190], aggregation scheduling [153, 180], asynchronous aggregation [29, 174], temporally weighted aggregation [121], controlled averaging algorithms [77], iterative round reduction [114, 148], and shuffled model aggregation [50]. These approaches aim to: (1) reduce communication cost and latency for better communication efficiency and scalability; (2) manage the device computation and energy resources to solve system heterogeneity and system performance issues, and (3) select high quality models for aggregation based on the model performance. Some researchers have proposed secure aggregation to solve data security and client device security issues [17, 19, 20].

Decentralised aggregation is a type of model aggregation that removes the central server from the federated learning systems and is mainly adopted to solve system reliability limitations. The decentralised aggregation can be realised through a peer-to-peer system [138, 144], one-hop neighbours collective learning [85], and **Online Push-Sum (OPS)** methods [59].

Training management is the second most proposed approach in the studies. It is used to address statistical heterogeneity, system heterogeneity, communication efficiency, client motivatability, and client device security issues. To deal with the statistical heterogeneity issue, researchers have proposed various training methods, such as the clustering of training data [23, 141], multi-stage training and fine-tuning of models [71], brokered learning [47], distributed multitask learning [36, 149], and edge computing [101]. The objectives are to increase the training data and reduce data skewness effect while maintaining a distributed manner. Furthermore, to address system heterogeneity issues, some methods balance the tasks and resources of the client nodes for the training process [39, 53, 89].

Another frequently proposed approach is the incentive mechanism, which is a solution to increase client motivatability. There is no obligation for data or device owners to join the training process if there are no benefits for them to contribute their data and resources. Incentive mechanisms can attract data and device owners to join the model training. Incentives can be given based on the amount of computation, communication, and energy resources provided [80, 165], the local model performance [107], the quality of the data provided [194], the honest behavior of client devices [130], and the dropout rate of a client node [125]. The proposed incentive mechanisms can be hosted by either a central server [75, 196] or a blockchain [79, 105, 210].

Resource management is introduced to control and optimize computation resources, bandwidth, and energy consumption of participating devices in a training process to address system heterogeneity [121, 168] and communication efficiency [4, 141]. The proposed approaches use control algorithms [168], reinforcement learning [118, 193], and edge computing methods [121] to optimise the resource usage and improve the system efficiency. To address communication efficiency, model

compression methods are utilised to reduce the data size and lower the communication cost that occurs during the model updates [22, 38, 82, 90, 142, 167]. Furthermore, model compression can also promote scalability as it is applicable to bandwidth limited and latency-sensitive scenarios [142].

Privacy preservation methods are introduced to maintain (1) data security: prevents information (model parameters or gradients) leakage to unauthorised parties, and (2) client devices security: prevents the system from being compromised by dishonest nodes. One well-cited method used for data security maintenance is differential privacy [139, 178], such as gaussian noise addition to the features before model updates [109]. For client device security maintenance, secure multiparty computations method such as homomorphic encryption is used together with local differential privacy method [139, 158, 178]. While homomorphic encryption only allows the central server to compute the global model homomorphically based on the encrypted local updates, the local differential privacy method protects client data privacy by adding noise to model parameter data sent by each client [158]. Security protection method is also proposed to solve the issues for both client device security [100, 102, 175] and data security [55, 198], which includes encrypting model parameters or gradients prior to exchanging the models between the server and client nodes.

Communication coordination methods are introduced to improve communication efficiency [65, 182], and system performance [34]. Specifically, communication techniques such as multi-channel random access communication [34] or over-the-air computation methods [6, 182] enable wireless federated training process to achieve faster convergence.

To address the model performance issues [64], statistical heterogeneity problems [161], system auditability limitations [164] and communication efficiency limitations [189], feature fusion or selection methods are used. Feature selection methods are used to improve the convergence rate [64] by only aggregating models with the selected features. Feature fusion is used to reduce the impact of non-IID data on the model performance [161]. Moreover, the feature fusion method reduces the dimension of data to be transferred to speed up the training process and increases the communication efficiency [189]. Reference [164] proposed a feature separation method that measures the importance level of the feature in the training process, used for system interpretation and auditing purposes.

Data provenance mechanisms are used to govern the data and information interactions between nodes. They are effective in preventing single-point-of-failures and adversarial attacks that intend to tamper the data. One way to implement a data provenance mechanism in a federated learning system is through blockchains. Blockchains record all the events instead of the raw data for audit and data provenance, and only permits authorised parties to access the information [105, 191]. Furthermore, blockchains are used for incentive provisions in References [14, 79, 80, 171], also for storing global model updates in Merkle trees [110].

Data augmentation methods are introduced to address data security [128, 155] and client device security issues [124]. The approaches use privacy-preserving generative adversarial network models to locally reproduce the data samples of all devices [68] for data release [155] and debugging processes [10]. These methods provide extra protection to shield the actual data from exposure. Furthermore, data augmentation methods are also effective in solving statistical heterogeneity issues through reproduction of an IID dataset for better training performance [68].

Auditing mechanisms are proposed as a solution for the lack of system auditability and client device security limitations. They are responsible for assessing the honest or semi-honest behavior of a client node during training and detecting any anomalies [10, 80, 201].

Some researchers have also introduced anomaly detection mechanisms [48, 94, 130], specifically to penalise adversarial or misbehaving nodes.

To properly assess the behavior of a federated learning system, researchers have proposed several evaluation mechanisms. The mechanisms intend to solve the system auditability and statistical heterogeneity issues. For instance, in Reference [170], a visualisation platform to illustrate the

Table 15. Summary of Component Designs

Sub-component	Client-based	Server-based	Both
Anomaly detector	—	3	—
Auditing mechanism	—	2	—
Data provenance	7	14	—
Client selector	—	1	—
Communication coordinator	—	6	—
Data augmentation	4	3	—
Encryption mechanism	3	—	12
Feature fusion mechanism	6	—	—
Incentive mechanism	2	9	—
Model aggregator	—	188	—
Model compressor	—	—	6
Model trainer	211	—	—
Privacy preservation mechanism	1	—	14
Resource manager	2	9	—
Training manager	1	9	—

federated learning system is presented. In Reference [21], a benchmarking platform is presented to realistically capture the characteristics of a federated scenario. Furthermore, Reference [61] introduces a method to measure the performance of the federated averaging algorithm.

Findings of RQ 3: How are the federated learning challenges being addressed?
Top 5 proposed approaches: The top 5 proposed approaches are model aggregation, training management, incentive mechanisms, privacy-preserving methods, and resource management. These approaches mainly aim to solve issues such as communication efficiency, statistical and system heterogeneity, client motivatability, and data privacy.
Remaining proposed approaches: A few papers worked on anomaly detection, auditing mechanisms, feature fusion/selection, evaluation mechanisms and data provenance.
With regard to the software development lifecycle, this question contributes to the *architecture design* phase where we summarise different approaches proposed to address the identified requirements of federated learning systems.

3.8 RQ 3.1: What are the Possible Components of Federated Learning Systems?

The motivation of RQ 3.1 is to identify the components in a federated learning system and their responsibilities. We classify the components of federated learning systems into two main categories: central server and client devices. A central server initiates and orchestrate the training process, whereas the client devices perform the actual model training [82, 113].

Apart from the central server and client devices, some studies added edge device to the system as an intermediate hardware layer between the central server and client devices. The edge device is introduced to increase the communication efficiency by reducing the training data sample size [72, 131] and increases the update speed [37, 101]. Table 15 summarises the number of mentions of each component. We classify them into client-based, server-based, or both to identify where these components are hosted. We can see that the model trainer is the most mentioned component for the client-based components, and model aggregator is mentioned the most in the server-based components. Furthermore, we notice that the software components that manage the system (e.g., anomaly detector, data provenance, communication coordinator, resource manager

and client selector) are mostly server-based while the software components that enhance the model performance (e.g., feature fusion, data augmentation) are mostly client-based. Last, two-way operating software components such as model encryption, privacy preservation, and model compressor exist in both clients and the server.

3.8.1 Central Server. Physically, the central servers are usually hosted on a local machine [88, 192], cloud server [56, 154, 177], mobile edge computing platform [121], edge gateway [146, 205], or base station [9, 65, 106, 182]. The central server is a hardware component that creates the first global model by randomly initialises the model parameters or gradient [27, 55, 88, 113, 114, 199]. Besides randomising the initial model, central servers can pre-train the global model, either using a self-generated sample dataset or a small amount of data collected from each client device [131, 132]. We can define this as the server-initialised model training setting. Note that not all federated learning systems initialise their global model on the central server. In decentralised federated learning, clients initialise the global model [14, 110, 143].

After global model initialisation, the central servers broadcast the global model that includes the model parameters or gradients to the participating client devices. The global model can be broadcasted to all the participating client devices every round [5, 65, 142, 206], or only to specific client devices, either randomly [32, 88, 114, 159, 182] or through selection based on the model training performance [131, 179] and the resources availability [38, 168]. Similarly, the trained local models are also collected from either all the participating client devices [167, 168, 198] or only from selected client devices [18, 111]. The collection of models can either be in an asynchronous [29, 64, 99, 174], or synchronous manner [42, 48]. Finally, the central server performs model aggregations when it receives all or a specific amount of updates, followed by the redistribution of the updated global model to the client devices. This entire process continues until convergence is achieved.

Apart from orchestrating the model parameters and gradients exchange, the central server also hosts other software components, such as encryption/decryption mechanisms for model encryption and decryption [11, 18, 111], and resource management mechanisms to optimise the resource consumption [9, 121]. Evaluation framework is proposed to evaluate the model and system performance [21, 61, 170], while the client and model selector is introduced to select appropriate clients for model training and select high quality models for global aggregation [38, 131, 168, 179]. Feature fusion mechanisms are proposed to combine essential model features and reduces communication cost [64, 161, 164, 189]. Incentive mechanisms are utilised to motivate the clients' participation rate [46, 78, 125, 126, 140, 165, 196, 210]. An anomaly detector is introduced to detect system anomaly [48], while the model compressor compresses the model to reduce its size [22, 38, 82, 142, 167]. Communication coordinator manages the multi-channel communication between the central server and client devices [5, 6, 27, 65, 147, 162, 182]. Last, auditing mechanisms audit the training processes [10, 14, 14, 201].

3.8.2 Client Devices. The client devices are the hardware component that conducts model training using the locally available datasets. First, each client device collects and pre-processes the local data (data cleaning, labeling, feature extraction, etc.). All client devices receive the initial global model and initiates the operations. The client devices decrypts and extracts the global model parameters. After that, they perform local model training. The received global model is also used for data inference and prediction by the client devices.

The local model training minimises the loss function and optimises the local model performance. Typically, the model is trained for multiple rounds and epochs [113], before being uploaded back to the central server for model aggregations. To reduce the number of communication rounds, Reference [52] proposed to perform local training on multiple local mini-batch of data. The method only communicates with the central server after the model achieved convergence.

After that, the client devices send the training results (model parameters or gradients) back to the central server. Before uploading, client devices evaluate the local model performance and only upload when an agreed level of performance is achieved [61]. The results are encrypted using the encryption mechanism before uploading to preserve the data security and prevent information leakage [11, 18, 111]. Furthermore, the model is compressed before uploaded to the central server to reduce communication cost [22, 38, 82, 142, 167]. In certain scenarios, not all devices are required to upload their results. Only the selected client devices are required to upload the results, depending on the selection criteria set by the central server. The criteria evaluates the available resources of the client devices and the model performance [9, 121, 193]. The client devices can host data augmentation mechanism [68, 155], and feature fusion mechanisms that correlate to the central server in the same federated learning systems. After the completion of one model training round, the training result is uploaded to the central server for global aggregation.

For decentralised federated learning systems, the client devices communicate among themselves without the orchestration of the central server. The removed central server is mostly replaced by a blockchain as a software component for model and information provenance [11, 14, 74, 79, 80, 105, 110, 171]. The blockchain is also responsible for incentive provision and differential private multiparty data model sharing. The initial model is created locally by each client device using local datasets. The models are updated using a consensus-based approach that enables devices to send model updates and receive gradients from neighbour nodes [85, 87, 143]. The client devices are connected through a peer-to-peer network [63, 138, 144]. Each device has the model update copies of all the client devices. After reaching consensus, all the client devices will conduct model training using the new gradient.

In cross-device settings, the system has a high client device population where each device is the data owner. In contrast, in the cross-silo setting, the client network is formed by several companies or organisations, regardless of the number of individual devices they own. The number of data silos is significantly smaller compared to the cross-device setting [73]. Therefore, the cross-device system creates models for large-scale distributed data on the same application [160], while the cross-silo system creates models to accommodate data that is heterogeneous in terms of its content and semantic in both features and sample space [160]. The data partitioning is also different. For the cross-device setting, the data is partitioned automatically by example (horizontal data partitioning). The data partitioning of cross-silo setting is fixed either by feature (vertical) or by example (horizontal) [73, 211].

Findings of RQ 3.1: What are the possible components of federated learning?
Mandatory components (clients): data collection, data preprocessing, feature engineering, model training, and inference.
Mandatory components (server): model aggregation, evaluation.
Optional components (clients): anomaly detection, model compression, auditing mechanisms, data augmentation, feature fusion/selection, security protection, privacy preservation, data provenance.
Optional components (server): advanced model aggregation, training management, incentive mechanism, resource management, communication coordination.
With regard to the software development lifecycle, this question contributes to the *architecture design* phase where we discuss the roles, responsibilities, and the interactions of the different components from an architecture design perspective.
**Mandatory components - Components that perform the main federated learning operations.*
**Optional components - Components that assist/enhance the federated learning operations.*

Table 16. Summary of Machine Learning Pipeline Phases

ML pipeline	Data collection	Data cleaning	Data labelling	Data augmentation	Feature engineering	Model training	Model evaluation	Model deployment	Model inference
Paper count	22	18	13	9	8	161	10	1	2

Table 17. Evaluation Approaches for Federated Learning

Evaluation methods	Application types	Paper count
Simulation (85%)	Image processing Others	99 98
Case study (7%)	Mobile device applications	11
	Healthcare	3
	Others	3
Both (1%)	Image processing	1
	Mobile device applications	1
No evaluation (7%)	—	15

3.9 RQ 3.2: What are the Phases of the Machine Learning Pipeline Covered by the Research Activities?

Table 16 presents a summary of machine learning pipeline phases covered by the studies. Notice that only phases that are specifically elaborated in the papers are included. The top 3 most mentioned machine learning phases are "model training" (161 mentions), followed by "data collection" (22 mentions) and "data cleaning" (18 mentions). These 3 stages of federated learning are mostly similar to the approaches in conventional machine learning systems. The key differences are the distributed model training tasks, decentralised data storage, and non-IID data distribution. Notice that only Google mentioned model inference and deployment, specifically for Google keyboard applications. The on-device inference supported by TensorFlow Lite is mentioned in References [57, 134], and Reference [185] mentioned that a model checkpoint from the server is used to build and deploy the on-device inference model. It uses the same featurisation flow that originally logged training examples on-device. However, the deployed model monitoring (e.g., dealing with performance degradation) and project management (e.g., model versioning) are not discussed in the existing studies. We infer that federated learning research is still in an early stage as most researchers focused on the data-processing and model training optimisation.

Findings of RQ 3.2:

What are the phases of the machine learning pipeline covered by the research activities?

Phases: The model training phase is most discussed. Only a few studies expressed data preprocessing, feature engineering, model evaluation, and only Google has discussions about model deployment (e.g., deployment strategies) and model inference. Model monitoring (e.g., dealing with performance degradation), and project management (e.g., model versioning) are not discussed in the existing studies. More studies are needed for the development of production-level federated learning systems.

With regard to the software development lifecycle, this question contributes to the *background understanding* phase where we identify the machine learning pipeline phases focused by the federated learning studies.

Table 18. Quality Attributes vs. Evaluation Metrics

Quality attributes vs Evaluation metrics	Communication efficiency	Model performance	Scalability	System performance	Statistical heterogeneity	System heterogeneity	Client motivatability	Data security	Client device security
Attack rate	—	—	—	1	—	—	—	1	4
Communication cost	58	—	3	2	—	—	—	—	—
Computation cost	—	—	—	42	—	—	—	—	—
Convergence rate	—	—	—	15	—	—	—	—	
Dropout ratio	1	—	—	2	—	2	—	1	—
Incentive rate	—	—	—	—	—	—	6	—	—
Model performance	1	253	—	1	—	—	2	—	—
Privacy loss	—	—	—	—	—	—	—	2	—
System running time	2	—	2	20	—	—	—	—	—
Qualitative evaluation	—	—	—	—	3	1	—	7	4

3.10 RQ 4: How are the Approaches Evaluated?

RQ 4 focuses on the evaluation approaches in the studies. We classify the evaluation approaches into two main groups: simulation and case study. For the simulation approach, image processing is the most common task, with 99 of 197 cases, whereas the most implemented use cases are applications on mobile devices (11 of 17 cases), such as word suggestion and human activity recognition.

Findings of RQ 4: How are the approaches evaluated?

Evaluation: Researchers mostly evaluate their federated learning approaches by simulation using privacy-sensitive scenarios. There are only a few real-world case studies, e.g., Google's mobile keyboard prediction.

With regard to the software development lifecycle, this question contributes to the *implementation and evaluation* phase where we explore the different methods to evaluate the federated learning approaches.

3.11 RQ 4.1: What are the Evaluation Metrics Used to Evaluate the Approaches?

Through RQ 4.1, we intend to identify the evaluation metrics for both qualitative and quantitative methods adopted by federated learning systems. We explain how each evaluation metric is used to assess the system and map these metrics to the quality attributes mentioned in RQ 2. The results are summarised in Table 18.

First, the communication efficiency is evaluated by communication cost, dropout ratio, model performance, and system running time. The communication cost is quantified by the communication rounds against the learning accuracy [32, 34, 103, 107, 114, 151, 167], the satisfying rate of communications [199], communication overhead versus number of clients [56, 181], the theoretical analysis of communication cost of data interchange between the server and clients [18, 150, 157], data transmission rate [175, 178], bandwidth [133, 193], and latency of communication [79]. The dropout ratios are measured by the computation overhead against dropout ratios [122, 175]. The

results are showcased as the comparison between communication overhead for different dropout rates [175], and the performance comparison against dropout rate [29, 96, 104].

Second, the model performance is measured by the training loss [25, 103, 151], AUC-ROC value [103, 105, 184], F1-score [13, 29, 43], root-mean-squared error [26, 148], cross-entropy [45, 174], precision [43, 186, 200], recall [43, 186, 200], prediction error [83, 149], mean absolute error [64], dice coefficient [145], and perplexity value [141].

Third, the system scalability is evaluated by communication cost and system running time. For the system running time evaluation, the results are presented as the total execution time of the training protocol (computation time and communication time) [104, 105, 117, 175, 198], the running time of different operation phases [198], the running time of each round against the number of client devices [18, 121, 130], and the model training time [171, 178].

The system performance is evaluated in multiple aspects, including system security (e.g., attack rate), scalability (e.g., communication and computation costs, dropout ratio), and system reliability (e.g., convergence rate, model performance, and system running time). The attack rate is measured as the proportion of attack targets that are incorrectly classified as the target label [48]. Essentially, researchers use this metric to evaluate how effective the defence mechanisms are [47, 117, 144, 201]. The types of attack are model poisoning attack, sybil attack, byzantine attack, and data reconstruction attack. Computation cost is the assessment of computation, storage, and energy resource usage of a system. The computation resources are quantified by the computation overhead against the number of client devices [55, 115, 122, 193], average computation time [72, 92, 187], computation throughput [9, 171], computation latency [79], computation utility, and the overhead of components [25, 196]. The storage resources are evaluated by the memory and storage overhead [42, 111, 122], and storage capacity [133]. The energy resources are calculated by the energy consumption for communication [46, 101], the energy consumption against computation time [92, 101, 146, 179, 187], and the energy consumption against training dataset size [209]. Last, the convergence rate is quantified by the accuracy versus the communication rounds, system running time, and training data epochs [15, 140, 189].

For statistical and system heterogeneity, qualitative analyses are conducted to verify if the proposed approaches have satisfied their purpose of addressing the limitations. The statistical heterogeneity is evaluated through formal verification such as model and equation proving [77, 116, 174]. System heterogeneity is evaluated through the devices dropout ratio due to the limited resource, and the formal verification through equation proving [116].

Client motivatability is measured by the incentive rate against different aspects of the system. The incentive rate is assessed by calculating the profit of the task publisher under different numbers of clients or accuracy levels [74], the average reward based on the model performance [204], and the relationship between the offered reward rate and the local accuracy over the communication cost [78, 125, 126]. From the studies collected, there is no mention of any specific form of rewards provided as incentives. However, cryptocurrencies such as Bitcoin or tokens that can be converted to actual money are common kinds of rewards.

Finally, both data security and client device security are measured by the attack rates and other respective qualitative evaluation metrics. Essentially, the data security analyses include analysis of encryption and verification process performance [16, 50, 56], the differential privacy achievement [105, 106], the effect of the removal of centralised trust [105], and the guarantee of shared data quality [105]. The client device security analyses are the performance of encryption and verification process [102, 175, 184], the confidentiality guarantee for gradients, the auditability of gradient collection and update, and the fairness guarantee for model training [171]. Also, the data security is measured by privacy loss, which evaluates the privacy-preserving level of the proposed method [204], derived from the differential average-case privacy [155].

Table 19. Comparison with Existing Reviews/Surveys on Federated Learning

Paper	Type	Time frames	Methodology	Scoping
This study	SLR	2016-2020	SLR guideline [81]	Software engineering perspective
Yang et al. (2019) [183]	Survey	2016-2018	Undefined	General overview
Kairouz et al. (2019) [73]	Survey	2016-2019	Undefined	General overview
Li et al. (2020) [93]	Survey	2016-2019	Customised	System view
Li et al. (2019) [95]	Survey	2016-2020	Undefined	General overview
Niknam et al. (2020) [120]	Review	2016-2019	Undefined	Wireless communications
Lim et al. (2020) [98]	Survey	2016-2020	Undefined	Mobile edge networks
Lyu et al. (2020) [108]	Survey	2016-2020	Undefined	Vulnerabilities
Xu and Wang (2019) [176]	Survey	2016-2019	Undefined	Healthcare informatics

Findings of RQ 4.1: What are the evaluation metrics used to evaluate the approaches?
Evaluation: Both quantitative and qualitative analysis are used to evaluate the federated learning system.
Quantitative metrics examples: Model performance, communication and computation cost, system running time, etc.
Qualitative metrics examples: Data security analysis on differential privacy achievement, Performance of encryption and verification process, confidentiality guarantee for gradients, the auditability of gradient collection and update, etc.
With regard to the software development lifecycle, this question contributes to the *implementation and evaluation* phase where we identify the different evaluation metrics used to assess the quality attributes of the federated learning systems.

3.12 Summary

We have presented all the findings and results extracted through each RQ. We summarise all the findings in a mind-map, as shown in Figure 9.

4 OPEN PROBLEMS AND FUTURE TRENDS IN FEDERATED LEARNING

In this section, we discuss the open problems and future research trends from the survey and review papers we collected to provide unbiased analyses (refer Table 19).

The findings are shown in Figure 8 and the detailed explanations are elaborated below:

- **Enterprise and industrial-level implementation.** Being at the early stage of research [73, 93, 95, 98, 176], the only mentions of enterprise federated learning systems are the cross-silo settings and some possible applications on real-world use cases listed in Table 10. The possible challenges are as follows:
  - **Machine learning pipeline:** Most existing studies in federated learning focus on the federated model training phase, without considering other machine learning pipeline stages (e.g., model deployment and monitoring) under the context of federated learning (e.g., new data lifecycles) [93, 95, 98, 176].
  - **Benchmark:** Benchmarking schemes are needed to evaluate the system development under real-world settings, with rich datasets and representative workload [73, 93, 95].

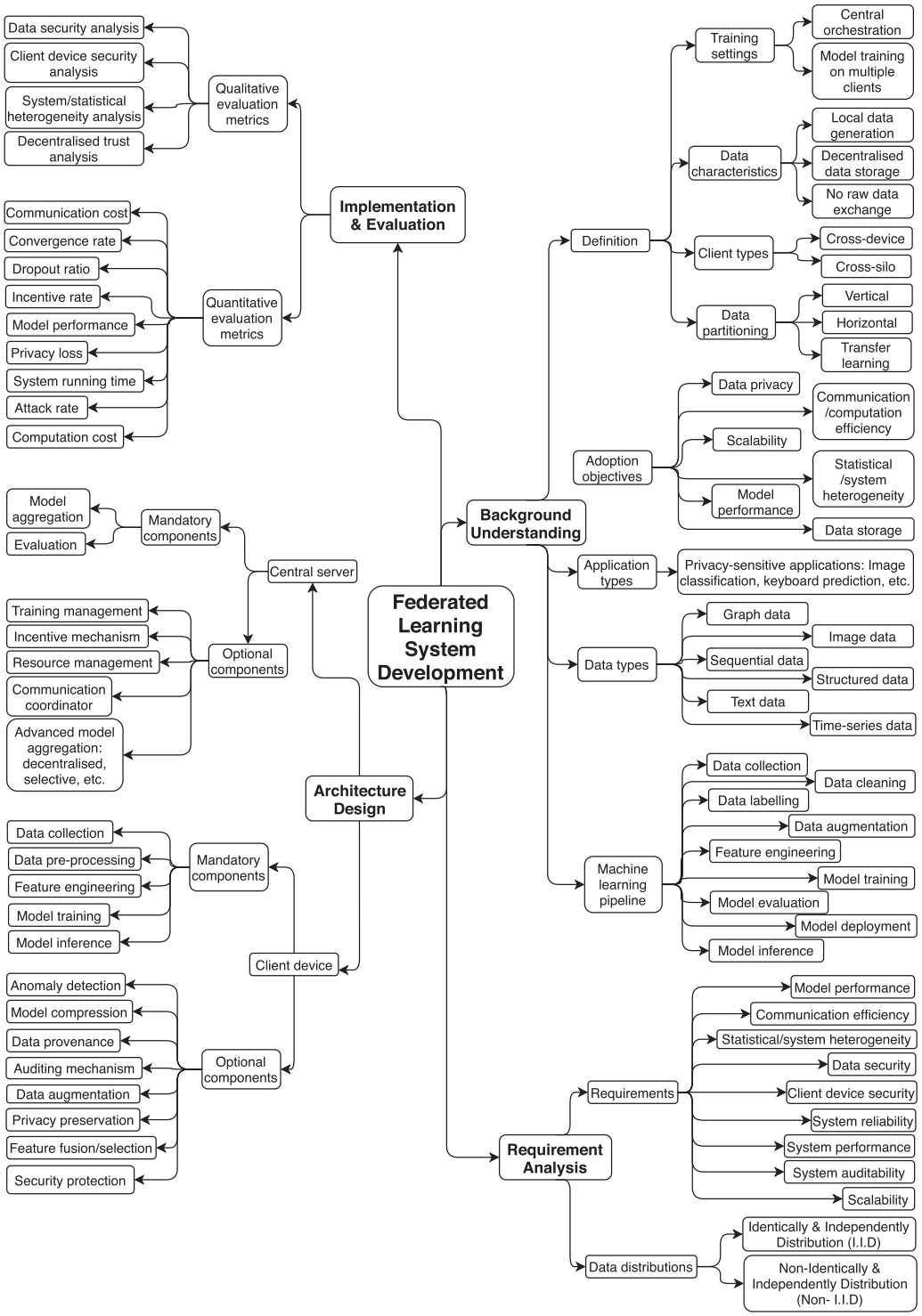

Fig. 9. Mind-map summary of the findings.

  - **Dealing with unlabeled client data:** In practice, the data generated on clients may be mislabeled or unlabeled [73, 93, 95, 98]. Some possible solutions are semi-supervised learning-based techniques, labeling the client data by learning the data label from other clients. However, these solutions may require dealing with the data privacy, heterogeneity, and scalability issues.
  - **Software architecture:** Federated learning still lacks systematic architecture design to guide methodical system development. A systematic architecture can provide design or algorithm alternatives for different system settings [93].
  - **Regulatory compliance:** The regulatory compliance issues for federated learning systems is under-explored [93] (e.g., whether data transfer limitations in GDPR is applied to model update transfer, ways to execute right-to-explainability to the global model, and whether the global model should be retrained if a client wants to quit). Machine learning and law enforcement communities are expected to cooperate to fill the gap between federated learning technology and the regulations in reality.
  - **Human-in-the-loop:** Domain experts are expected to be involved in the federated learning process to provide professional guidance as end-users tend to trust the expert's judgement more than the inference by machine learning algorithms. [176].
- **Advanced privacy preservation methods:** More advanced privacy preservation methods are needed as the existing solutions still can reveal sensitive information [73, 93, 95, 98, 108, 120].
  - **Tradeoffs between data privacy and model system performance:** The current approaches (e.g., differential privacy and collaborative training) sacrifice the model performance, and require significant computation cost [98, 108, 120]. Hence, the design of federated learning systems needs to balance the tradeoffs between data privacy and model/system performance.
  - **Granular privacy protection:** In reality, privacy requirements may differ across clients or across data samples on a single client. Therefore, it is necessary to protect data privacy in a more granular manner. Privacy heterogeneity should be considered in the design of federated learning systems for different privacy requirements (e.g., client device-specific or sample data specific). One direction of future work is extending differential privacy with granular privacy restriction (e.g., heterogeneous differential privacy) [93, 95, 108].
  - **Sharing of less sensitive model data:** Devices should only share less sensitive model data (e.g., inference results or signSGD), and this type of approaches may be considered in future work [73, 108].
- **Improving system and model performance.** There are still some performance issues regarding federated learning systems, mainly on resource allocation (e.g., communication, computation, and energy efficiency). Moreover, model performance improvement through the non-algorithm or non-gradient optimisation approach (e.g., promoting more participants, the extension of the federated model training method) is also another future research trend.
  - **Handling of client dropouts:** In practice, participating clients may drop out from the training process due to energy constraints or network connectivity issues [73, 93, 98]. A large number of client dropouts can significantly degrade the model performance. Many of the existing studies do not consider the situation when the number of participating clients changes (i.e., departures or entries of clients). Hence, the design of a federated learning system should support the dynamic scheduling of model updates to tolerate client dropouts. Furthermore, new algorithms are needed to deal with the scenarios where

only a small number of clients are left in the training rounds. The learning coordinator should provide stable network connections to the clients to avoid dropouts.

- **Advanced incentive mechanisms:** Without a well-designed incentive mechanism, potential clients may be reluctant to join the training process, which will discourage the adoption of federated learning [73, 93, 176]. In most of the current designs, the model owner pays for the participating clients based on some metrics (e.g., number of participating rounds or data size), which might not be effective in evaluating the incentive provision.
- **Model markets:** A possible solution proposed to promote federated learning applications is the model market [93]. One can perform model prediction using the model purchased from the model market. In addition, the developed model can be listed on the model market with additional information (e.g., task, domain) for federated transfer learning.
- **Combined algorithms for communication reduction:** The method to combine different communication reduction techniques (combining model compression technique with local updating) is an interesting direction to further improve the system's communication efficiency (e.g., optimise the size of model updates and communication instances) [73, 95, 98]. However, the feasibility of such a combination is still under-explored. In addition, the tradeoffs between model performance and communication efficiency are needed to be further examined (e.g., how to manage the tradeoffs under the change of training settings?).
- **Asynchronous federated learning:** Synchronous federated learning may have efficiency issues caused by stragglers.
  Asynchronous federated learning has been considered as a more practical solution, even allowing clients to participate in the training halfway [73, 95, 98]. However, new asynchronous algorithms are still under explored to provide convergence guarantee.
- **Statistical heterogeneity quantification:** The quantification of the statistical heterogeneity into metrics (e.g., local dissimilarity) is needed to help improve the model performance for non-IID condition [95, 98]. However, the metrics can hardly be calculated before training. One important future direction is the design of efficient algorithms to calculate the degree of statistical heterogeneity that is useful for model optimisations.
- **Multi-task learning:** Most of the existing studies focus on training the same model on multiple clients. One interesting direction is to explore how to apply federated learning to train different models in the federated networks [73, 108].
- **Decentralised learning:** As mentioned above, decentralised learning does not require a central server in the system [73, 108], which prevents single-point-of-failure. Hence, it would be interesting to explore if there is any new attack or whether federated learning security issues still exist in this setting.
- **One/few shot learning:** Federated learning that executes less training iterations, such as one/few-shot learning, has been recently discussed for federated learning [73, 95]. However, more theoretical and empirical studies are needed in this direction.
- **Federated transfer learning:** For federated transfer learning, there are only 2 domains or data parties are assumed in most of the current literature. Therefore, expanding federated transfer learning to multiple domains, or data parties is an open problem [73].

5 THREATS TO VALIDITY

We identified the threats to validity that might influence the outcomes of our research. First, publication bias exists as most studies have positive results rather than negative results. While

studies with positive results are much appealing to be published in comparison with studies with no or negative results, a tendency towards certain outcomes might leads to biased conclusions. The second threat is the exclusion of the studies that focus on pure algorithm improvements. The exclusion of respective studies may affect the completeness of this research as some discussion on the model performance and data heterogeneity issues might be relevant. The third threat is the incomplete search strings in the automatic search. We included all the possible supplementary terms related to federated learning while excluding keywords that return conventional machine learning publications. However, the search terms in the search strings may still be insufficient to search all the relevant work related to federated learning research topics. The fourth threat is the exclusion of ArXiv and Google Scholar papers that are not cited by peer-reviewed papers. The papers from these sources are not peer-reviewed, and we cannot guarantee the quality of these research works. However, we want to collect as many state-of-the-art studies in federated learning as possible. To maintain the quality of the search pool, we only include papers that are cited by peer-reviewed papers. The fifth threat is the time span of research works included in this systematic literature review. We only included papers published from 2016.01.01 to 2020.01.31. Since the data in 2020 does not represent the research trend of the entire year, we only include works from 2016 to 2019 for the research trend analysis (refer to Figure 6). However, the studies collected in only January 2020 is equally significant to those from the previous years for identifying challenges and solutions. Hence, we keep the findings from the papers published in 2020 for the remaining discussions. The sixth threat is the study selection bias. To avoid the study selection bias between the two researchers, the cross-validation of the results from the pilot study is performed by the two independent researchers prior to the update of search terms and inclusion/exclusion criteria. The mutual agreement between the two independent researchers on the selection of papers is required. When a dispute on the decision occurs, the third researcher is consulted. Last, there might be bias in data collection and analysis due to different background and experience of the researchers.

6 RELATED WORK

According to the protocol, we collected all the relevant surveys and reviews. There are seven surveys and one review that studied the federated learning topic. To the best of our knowledge, there is still no systematic literature review conducted on federated learning.

Li et al. [93] propose a federated learning building blocks taxonomy that classifies the federated learning system into 6 aspects: data partitioning, machine learning model, privacy mechanism, communication architecture, the scale of the federation, and the motivation of federation. Kairouz et al. [73] present a general survey on the advancement in research trends and the open problems suggested by researchers. Moreover, the paper covered detailed definitions of federated learning system components and different types of federated learning systems variations. Li et al. [95] present a review on federated learning's core challenges of federated learning in terms of communication efficiency, privacy, and future research directions. Surveys on federated learning systems for specific research domains are also conducted. Niknam et al. [120] review federated learning in the context of wireless communications. The survey mainly investigates the data security and privacy challenges, algorithm challenges, and wireless setting challenges of federated learning systems. Lim et al. [98] discuss federated learning papers in the mobile edge network. Lyu et al.[108] focus on the security threats and vulnerability challenges in federated learning systems whereas Xu and Wang [176] explore the healthcare and medical informatics domain. A review of federated learning that focuses on data privacy and security aspects was conducted by Reference [183].

The comparisons of each survey and review papers with our systematic literature review are summarised in Table 19. We compare our work with the existing works in four aspects: (1) Time frames: our review on the state-of-the-art is the most contemporary as it is the most up-to-date

review. (2) Methodology: We followed Kitchenham's standard guideline [81] to conduct this systematic literature review. Most of the existing works have no clear methodology, where information is collected and interpreted with subjective summaries of findings that may be subject to bias. (3) Comprehensiveness: the number of papers analysed in our work is higher than the existing reviews or surveys as we screened through relevant journals and conferences paper-by-paper. (4) Analysis: We provided two more detailed reviews on federated learning approaches, including (a) the context of studies (e.g., publication venues, year, type of evaluation metrics, method, and dataset used) for the state-of-the-art research identification, and (b) the data synthesis of the findings through the lifecycle of federated learning systems.

7 CONCLUSION

Federated learning has attracted a broad range of interests from academia and industry. We performed a systematic literature review on federated learning from the software engineering perspective with 231 primary studies. The results show that most of the known motivations for using federated learning appear to be also the most studied research challenges in federated learning. To tackle the challenges, the top five proposed approaches are model aggregation, training management, incentive mechanism, privacy preservation, and resource management. The research findings provide clear viewpoints on federated learning system development for production adoption. Finally, this article sheds some light on the future research trends of federated learning and encourages researchers to extend and advance their current work.

REFERENCES

[1] ISO/IEC 25012. 2008. Software engineering–software product quality requirements and evaluation (SQuaRE)–data quality model.

[2] ISO/IEC 25010. 2011. Software engineering–software product quality requirements and evaluation (SQuaRE)–system and software quality models.

[3] General Data Protection Regulation. 2018. EU data protection rules. 1821–1834. Retrieved from https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en.

[4] Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. 2019. Hierarchical federated learning across heterogeneous cellular networks. Retrieved from https://arXiv:1909.02362.

[5] J. Ahn, O. Simeone, and J. Kang. 2019. Wireless federated distillation for distributed edge learning with heterogeneous data. In *Proceedings of IEEE PIMRC*. 1–6.

[6] Mohammad Mohammadi Amiri and Deniz Gunduz. 2019. Federated learning over wireless fading channels. Retrieved from https://arXiv:1907.09769.

[7] Mohammad Mohammadi Amiri, Deniz Gunduz, Sanjeev R. Kulkarni, and H. Vincent Poor. 2020. Update Aware Device Scheduling for Federated Learning at the Wireless Edge. Retrieved from https://arxiv:cs.IT/2001.10402.

[8] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, and Antonio Ferrara. 2019. *Towards Effective Device-aware Federated Learning*. Springer International Publishing, Cham, 477–491.

[9] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L. Wang. 2019. Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach. *IEEE Wireless Commun. Lett.* 8, 5 (Oct. 2019), 1345–1348.

[10] Sean Augenstein, H. Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, and Blaise Aguera y Arcas. 2019. Generative Models for Effective ML on Private, Decentralized Datasets. Retrieved from https://arxiv:cs.LG/1911.06679.

[11] Sana Awan, Fengjun Li, Bo Luo, and Mei Liu. 2019. Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain. In *Proceedings of SIGSAC'19*. ACM, London, UK, 2561–2563.

[12] U. M. Avodji, S. Gambs, and A. Martin. 2019. IOTFLA: A secured and privacy-preserving smart home architecture implementing federated learning. In *Proceedings of IEEE SPW*. 175–180.

[13] Evita Bakopoulou, Balint Tillman, and Athina Markopoulou. 2019. A federated learning approach for mobile packet classification. Retrieved from https://arXiv:1907.13113.

[14] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu. 2019. FLChain: A blockchain for auditable federated learning with trust and incentive. In *Proceedings of BIGCOM*. 151–159.

[15] Daniel Benditkis, Aviv Keren, Liron Mor-Yosef, Tomer Avidor, Neta Shoham, and Nadav Tal-Israel. 2019. Distributed deep neural network training on edge devices. In *Proceedings of SEC'19*. ACM, Arlington, VA, 304–306.
[16] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. 2018. Protection against reconstruction and its applications in private federated learning. Retrieved from https://arXiv:1812.00984.
[17] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H. Brendan McMahan et al. 2019. Towards federated learning at scale: System design. Retrieved from https://arXiv:1902.01046.
[18] 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 SIGSAC'17*. ACM, Dallas, TX, 1175–1191.
[19] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical secure aggregation for federated learning on user-held data. Retrieved from https://arXiv:1611.04482.
[20] Keith Bonawitz, Fariborz Salehi, Jakub Konečnỳ, Brendan McMahan, and Marco Gruteser. 2019. Federated learning with autotuned communication-efficient secure aggregation. Retrieved from https://arXiv:1912.00131.
[21] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Kone, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. LEAF: A Benchmark for Federated Settings. Retrieved from https://arxiv:cs.LG/1812.01097.
[22] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. Retrieved from https://arXiv:1812.07210.
[23] Sebastian Caldas, Virginia Smith, and Ameet Talwalkar. 2018. Federated kernelized multi-task learning. In *Proceedings of SysML'18*.
[24] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2019. Secure Federated Matrix Factorization. Retrieved from https://arxiv:cs.CR/1906.05108.
[25] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated Meta-Learning with Fast Convergence and Efficient Communication. Retrieved from https://arxiv:cs.LG/1802.07876.
[26] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin. 2020. Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks. *IEEE Trans. Wireless Commun.* 19, 1 (Jan. 2020), 177–191.
[27] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H. Vincent Poor, and Shuguang Cui. 2019. A joint learning and communications framework for federated learning over wireless networks. Retrieved from https://arXiv:1909.07972.
[28] Xiangyi Chen, Tiancong Chen, Haoran Sun, Zhiwei Steven Wu, and Mingyi Hong. 2019. Distributed Training with Heterogeneous Data: Bridging Median- and Mean-based Algorithms. Retrieved from https://arxiv:cs.LG/1906.01736.
[29] Yujing Chen, Yue Ning, and Huzefa Rangwala. 2019. Asynchronous online federated learning for edge devices. Retrieved from https://arXiv:1911.02134.
[30] Yudong Chen, Lili Su, and Jiaming Xu. 2018. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *SIGMETRICS Perform. Eval. Rev.* 46, 1 (2018), 96.
[31] Yang Chen, Xiaoyan Sun, and Yaochu Jin. 2019. Communication-efficient Federated Deep Learning with Asynchronous Model Update and Temporally Weighted Aggregation. Retrieved from https://arxiv:cs.LG/1903.07424.
[32] Y. Chen, X. Sun, and Y. Jin. 2019. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* (2019), 1–10.
[33] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. 2019. SecureBoost: A lossless federated learning framework. Retrieved from https://arXiv:1901.08755.
[34] J. Choi and S. R. Pokhrel. 2019. Federated learning with multichannel ALOHA. *IEEE Wireless Commun. Lett.* (2019).
[35] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2019. Differential Privacy-enabled Federated Learning for Sensitive Health Data. Retrieved from https://arxiv:cs.LG/1910.02578.
[36] Luca Corinzia and Joachim M. Buhmann. 2019. Variational federated multi-task learning. Retrieved from https://arXiv:1906.06268.
[37] Harshit Daga, Patrick K. Nicholson, Ada Gavrilovska, and Diego Lugones. 2019. Cartel: A system for collaborative transfer learning at the edge. In *Proceedings of SoCC'19*. ACM, 25–37.
[38] K. Deng, Z. Chen, S. Zhang, C. Gong, and J. Zhu. 2019. Content compression coding for federated learning. In *Proceedings of WCSP'19*. 1–6.
[39] Canh Dinh, Nguyen H Tran, Minh NH Nguyen, Choong Seon Hong, Wei Bao, Albert Zomaya, and Vincent Gramoli. 2019. Federated learning over wireless networks: Convergence analysis and resource allocation. Retrieved from https://arXiv:1910.13067.

[40] R. Doku, D. B. Rawat, and C. Liu. 2019. Towards federated learning approach to determine data relevance in big data. In *Proceedings of IEEE IRI*. 184–192.
[41] Wei Du, Xiao Zeng, Ming Yan, and Mi Zhang. 2018. Efficient federated learning via variational dropout. Openreview.net, 1–12. Retrieved from https://openreview.net/forum?id=BkeAf2CqY7.
[42] Moming Duan. 2019. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. Retrieved from https://arXiv:1907.01132.
[43] S. Duan, D. Zhang, Y. Wang, L. Li, and Y. Zhang. 2019. JointRec: A deep learning-based joint cloud video recommendation framework for mobile IoTs. *IEEE IoT-J.* 7, 3 (2019), 1655–1666.
[44] Amin Fadaeddini, Babak Majidi, and Mohammad Eshghi. 2019. *Privacy Preserved Decentralized Deep Learning: A Blockchain-based Solution for Secure AI-driven Enterprise.* Springer International Publishing, Cham, 32–40.
[45] Zipei Fan, Xuan Song, Renhe Jiang, Quanjun Chen, and Ryosuke Shibasaki. 2019. Decentralized attention-based personalized human mobility prediction. *Proc. IMWUT* 3, 4 (2019), Article 133.
[46] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y. Liang. 2019. Joint service pricing and cooperative relay communication for federated learning. In *Proceedings of IEEE GreenCom.* 815–820.
[47] Clement Fung, Jamie Koerner, Stewart Grant, and Ivan Beschastnikh. 2018. Dancing in the Dark: Private Multi-Party Machine Learning in an Untrusted Setting. Retrieved from https://arxiv:cs.CR/1811.09712.
[48] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2018. Mitigating sybils in federated learning poisoning. Retrieved from https://arXiv:1808.04866.
[49] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. Retrieved from https://arXiv:1712.07557.
[50] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. 2019. Scalable and Differentially Private Distributed Aggregation in the Shuffled Model. Retrieved from https://arxiv:cs.LG/1906.08320.
[51] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. 2019. Robust federated learning in a heterogeneous environment. Retrieved from https://arXiv:1906.06629.
[52] Neel Guha, Ameet Talwlkar, and Virginia Smith. 2019. One-shot federated learning. Retrieved from https://arXiv:1902.11175.
[53] Pengchao Han, Shiqiang Wang, and Kin K. Leung. 2020. Adaptive Gradient Sparsification for Efficient Federated Learning: An Online Learning Approach. Retrieved from https://arxiv:cs.LG/2001.04756.
[54] Yufei Han and Xiangliang Zhang. 2019. Robust federated training via collaborative machine teaching using trusted instances. Retrieved from https://arXiv:1905.02941.
[55] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu. 2020. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Trans. Industr. Inform.* 16, 10 (2020), 6532–6542.
[56] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang. 2019. Towards efficient and privacy-preserving federated deep learning. In *Proceedings of ICC'19.* 1–6.
[57] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. Retrieved from https://arXiv:1811.03604.
[58] 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. Retrieved from https://arXiv:1711.10677.
[59] Chaoyang He, Conghui Tan, Hanlin Tang, Shuang Qiu, and Ji Liu. 2019. Central server free federated learning over single-sided trust social networks. Retrieved from https://arXiv:1910.04956.
[60] X. He, Q. Ling, and T. Chen. 2019. Byzantine-robust stochastic gradient descent for distributed low-rank matrix completion. *Proceedings of IEEE DSW.* 322–326.
[61] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. Retrieved from https://arxiv:cs.LG/1909.06335.
[62] B. Hu, Y. Gao, L. Liu, and H. Ma. 2018. Federated region-learning: An edge computing based framework for urban environment sensing. In *Proceedings of IEEE GLOBECOM.* 1–7.
[63] Chenghao Hu, Jingyan Jiang, and Zhi Wang. 2019. Decentralized federated learning: A segmented gossip approach. Retrieved from https://arXiv:1908.07782.
[64] Yao Hu, Xiaoyan Sun, Yang Chen, and Zishuai Lu. 2019. *Model and Feature Aggregation-based Federated Learning for Multi-sensor Time Series Trend Following.* Springer International Publishing, Cham, 233–246.
[65] S. Hua, K. Yang, and Y. Shi. 2019. On-device federated learning via second-order optimization with over-the-air computation. In *Proceedings of IEEE VTC'19.* 1–5.
[66] Li Huang, Andrew L. Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. 2019. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *J. Biomed. Inform.* 99 (2019), 103291.

[67] Amir Jalalirad, Marco Scavuzzo, Catalin Capota, and Michael Sprague. 2019. A simple and efficient federated recommender system. In *Proceedings of BDCAT'19*. ACM, Auckland, New Zealand, 53–58.

[68] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. Retrieved from https://arXiv:1811.11479.

[69] Shaoxiong Ji, Guodong Long, Shirui Pan, Tianqing Zhu, Jing Jiang, and Sen Wang. 2019. *Detecting Suicidal Ideation with Data Protection in Online Communities*. Springer International Publishing, Cham, 225–229.

[70] Di Jiang, Yuanfeng Song, Yongxin Tong, Xueyang Wu, Weiwei Zhao, Qian Xu, and Qiang Yang. 2019. Federated topic modeling. In *Proceedings of CIKM'19*. ACM, 1071–1080.

[71] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. 2019. Improving federated learning personalization via model agnostic meta learning. Retrieved from https://arXiv:1909.12488.

[72] Yuang Jiang, Shiqiang Wang, Bong Jun Ko, Wei-Han Lee, and Leandros Tassiulas. 2019. Model pruning enables efficient federated learning on edge devices. Retrieved from https://arXiv:1909.12326.

[73] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurlien 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, Adri Gascon, 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 Ozgür, 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. Retrieved from https://arxiv:cs.LG/1912.04977.

[74] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang. 2019. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE IoT-J.* 6, 6 (Dec. 2019), 10700–10714.

[75] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim. 2019. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In *Proceedings of IEEE APWCS*. 1–5.

[76] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani. 2020. Reliable federated learning for mobile networks. *IEEE Wirel. Commun.* 27, 2 (2020), 72–80.

[77] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2019. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. Retrieved from https://arXiv:1910.06378.

[78] Latif U. Khan, Nguyen H. Tran, Shashi Raj Pandey, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong. 2019. Federated learning for edge networks: Resource optimization and incentive mechanism. Retrieved from https://arXiv:1911.05642.

[79] H. Kim, J. Park, M. Bennis, and S. Kim. 2019. Blockchained on-device federated learning. *IEEE Commun. Lett.* 24, 6 (2019), 1279–1283.

[80] Y. J. Kim and C. S. Hong. 2019. Blockchain-based node-aware dynamic weighting methods for improving federated learning performance. In *Proceedings of APNOMS*. 1–4.

[81] B. Kitchenham and S. Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University.

[82] Jakub Konečnỳ, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. Retrieved from https://arXiv:1610.05492.

[83] Jakub Kone, H. Brendan McMahan, Daniel Ramage, and Peter Richtrik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. Retrieved from https://arxiv:cs.LG/1610.02527.

[84] Antti Koskela and Antti Honkela. 2019. Learning rate adaptation for federated and differentially private learning. *stat* 1050 (2019), 31.

[85] Anusha Lalitha, Osman Cihan Kilinc, Tara Javidi, and Farinaz Koushanfar. 2019. Peer-to-peer federated learning on graphs. Retrieved from https://arXiv:1901.11173.

[86] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. 2018. Fully decentralized federated learning. In *Proceedings of NeurIPS*.

[87] Anusha Lalitha, Xinghan Wang, Osman Kilinc, Yongxi Lu, Tara Javidi, and Farinaz Koushanfar. 2019. Decentralized Bayesian Learning over Graphs. Retrieved from https://arxiv:stat.ML/1905.10466.

[88] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau. 2019. Federated learning for keyword spotting. In *Proceedings of ICASSP*. 6341–6345.

[89] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous federated learning via model distillation. Retrieved from https://arXiv:1910.03581.

[90] Hongyu Li and Tianqi Han. 2019. An End-to-End Encrypted Neural Network for Gradient Updates Transmission in Federated Learning. Retrieved from https://arxiv:cs.LG/1908.08340.
[91] Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. 2019. Differentially Private Meta-learning. Retrieved from https://arxiv:cs.LG/1909.05830.
[92] L. Li, H. Xiong, Z. Guo, J. Wang, and C. Xu. 2019. SmartPC: Hierarchical pace control in real-time federated learning system. In *Proceedings of IEEE RTSS*. 406–418.
[93] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, and Bingsheng He. 2019. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. Retrieved from https://arxiv:cs.LG/1907.09693.
[94] Suyi Li, Yong Cheng, Yang Liu, Wei Wang, and Tianjian Chen. 2019. Abnormal client behavior detection in federated learning. Retrieved from https://arXiv:1910.09933.
[95] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* 37, 3 (May 2020), 50–60.
[96] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated Optimization in Heterogeneous Networks. Retrieved from https://arxiv:cs.LG/1812.06127.
[97] Tian Li, Maziar Sanjabi, and Virginia Smith. 2019. Fair resource allocation in federated learning. Retrieved from https://arXiv:1905.10497.
[98] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2019. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. Retrieved from https://arxiv:cs.NI/1909.11875.
[99] B. Liu, L. Wang, and M. Liu. 2019. Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems. *IEEE RA-L* 4, 4 (Oct 2019), 4555–4562.
[100] Changchang Liu, Supriyo Chakraborty, and Dinesh Verma. 2019. Secure model fusion for distributed learning using partial homomorphic encryption. In *Policy-Based Autonomic Data Governance*, Seraphin Calo, Elisa Bertino, and Dinesh Verma (Eds.). Springer International Publishing, Cham, 154–179.
[101] Lumin Liu, Jun Zhang, S. H. Song, and Khaled Ben Letaief. 2019. Client-edge-cloud hierarchical federated learning. Retrieved from https://arxiv.org/abs/1905.06641.
[102] Yang Liu, Tianjian Chen, and Qiang Yang. 2018. Secure Federated Transfer Learning. Retrieved from https://arxiv:cs.LG/1812.03337.
[103] Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. 2019. A communication efficient vertical federated learning framework. Retrieved from https://arXiv:1912.11187.
[104] Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, Surya Nepal, and Robert Deng. 2019. Boosting Privately: Privacy-Preserving Federated Extreme Boosting for Mobile Crowdsensing. Retrieved from https://arxiv:cs.CR/1907.10218.
[105] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang. 2019. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Trans. Industr. Inform.* 16, 6 (2019), 4177–4186.
[106] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang. 2019. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Trans. Industr. Inform.* 16, 3 (2019), 2134–2143.
[107] S. Lugan, P. Desbordes, E. Brion, L. X. Ramos Tormo, A. Legay, and B. Macq. 2019. Secure architectures implementing trusted coalitions for blockchained distributed learning (TCLearn). *IEEE Access* 7 (2019), 181789–181799.
[108] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to Federated Learning: A Survey. Retrieved from https://arxiv:cs.CR/2003.02133.
[109] Jing Ma, Qiuchen Zhang, Jian Lou, Joyce C. Ho, Li Xiong, and Xiaoqian Jiang. 2019. Privacy-preserving tensor factorization for collaborative health data analysis. In *Proceedings of CIKM'19*. ACM, 1291–1300.
[110] U. Majeed and C. S. Hong. 2019. FLchain: Federated learning via MEC-enabled blockchain network. In *Proceedings of APNOMS*. 1–4.
[111] Kalikinkar Mandal and Guang Gong. 2019. PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks. In *Proceedings of SIGSAC'19*. ACM, 57–68.
[112] I. Martinez, S. Francis, and A. S. Hafid. 2019. Record and reward federated learning contributions with blockchain. In *Proceedings of CyberC*. 50–57.
[113] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. Retrieved from https://arxiv:cs.LG/1602.05629.
[114] J. Mills, J. Hu, and G. Min. 2019. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE IoT-J* 7, 7 (2019), 5986–5994.
[115] Fan Mo and Hamed Haddadi. 2019. Efficient and private federated learning using TEE. Retrieved from https://eurosys2019.org/wp-content/uploads/2019/03/eurosys19posters-abstract66.pdf.
[116] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic federated learning. Retrieved from https://arXiv:1902.00146.

[117] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae. 2020. Federated learning-based cognitive detection of jamming attack in flying ad-hoc network. *IEEE Access* 8 (2020), 4338–4350.
[118] C. Nadiger, A. Kumar, and S. Abdelhak. 2019. Federated reinforcement learning for fast personalization. In *Proceedings of IEEE AIKE*. 123–127.
[119] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. Sadeghi. 2019. DÏoT: A federated self-learning anomaly detection system for IoT. In *Proceedings of IEEE ICDCS*. 756–767.
[120] Solmaz Niknam, Harpreet S. Dhillon, and Jeffery H. Reed. 2019. Federated Learning for Wireless Communications: Motivation, Opportunities and Challenges. Retrieved from https://arxiv:eess.SP/1908.06847.
[121] T. Nishio and R. Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *Proceedings of ICC'19*. 1–7.
[122] Chaoyue Niu, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, Chengfei Lv, Zhihua Wu, and Guihai Chen. 2019. Secure federated submodel learning. Retrieved from https://arXiv:1911.02254.
[123] 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. Retrieved from https://arXiv:1803.04035.
[124] Tribhuvanesh Orekondy, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz. 2018. Gradient-Leaks: Understanding and Controlling Deanonymization in Federated Learning. Retrieved from https://arxiv:cs.CR/1805.05838.
[125] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, Z. Han, and C. S. Hong. 2019. Incentivize to build: A crowdsourcing framework for federated learning. In *Proceedings of IEEE GLOBECOM*. 1–6.
[126] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong. 2020. A crowdsourcing framework for on-device federated learning. *IEEE Trans. Wireless Commun.* 19, 5 (2020), 3241–3256.
[127] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. 2019. Federated Adversarial Domain Adaptation. Retrieved from https://arxiv:cs.CV/1911.02054.
[128] Daniel Peterson, Pallika Kanani, and Virendra J. Marathe. 2019. Private federated learning with domain adaptation. Retrieved from https://arXiv:1912.06733.
[129] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. 2019. Robust aggregation for federated learning. Retrieved from https://arXiv:1912.13445.
[130] Davy Preuveneers, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth Ilie-Zudor. 2018. Chained anomaly detection models for federated learning: An intrusion detection case study. *Appl. Sci.* 8, 12 (2018), 2663.
[131] J. Qian, S. P. Gochhayat, and L. K. Hansen. 2019. Distributed active learning strategies on edge computing. In *Proceedings of IEEE CSCloud/EdgeCom*. 221–226.
[132] Jia Qian, Sayantan Sengupta, and Lars Kai Hansen. 2019. Active Learning Solution on Distributed Edge Computing. Retrieved from https://arxiv:cs.DC/1906.10718.
[133] Yongfeng Qian, Long Hu, Jing Chen, Xin Guan, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi. 2019. Privacy-aware service placement for mobile edge computing via federated learning. *Info. Sci.* 505 (2019), 562–570.
[134] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. 2019. Federated learning for emoji prediction in a mobile keyboard. Retrieved from https://arXiv:1906.04329.
[135] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2019. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. Retrieved from https://arXiv:1909.13014.
[136] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang. 2019. Federated learning-based computation offloading optimization in edge computing-supported Internet of Things. *IEEE Access* 7 (2019), 69194–69201.
[137] Jinke Ren, Guanding Yu, and Guangyao Ding. 2019. Accelerating DNN Training in Wireless Federated Edge Learning System. Retrieved from https://arxiv:cs.LG/1905.09712.
[138] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 2019. BrainTorrent: A peer-to-peer environment for decentralized federated learning. Retrieved from https://arXiv:1905.06731.
[139] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A generic framework for privacy preserving deep learning. Retrieved from https://arxiv:cs.LG/1811.04017.
[140] Y. Sarikaya and O. Ercetin. 2019. Motivating workers in federated learning: A stackelberg game perspective. *IEEE Netw. Lett.* 2, 1 (2019), 23–27.
[141] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2019. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints. Retrieved from https://arXiv:1910.01991.
[142] F. Sattler, S. Wiedemann, K. Muller, and W. Samek. 2019. Robust and communication-efficient federated learning from non-i.i.d. Data. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 9 (2019), 3400–3413.

[143] S. Savazzi, M. Nicoli, and V. Rampa. 2020. Federated learning with cooperating devices: A consensus approach for massive IoT networks. *IEEE IoT-J* 7, 5 (2020), 4641–4654.
[144] Muhammad Shayan, Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2018. Biscotti: A Ledger for Private and Secure Peer-to-Peer Machine Learning. Retrieved from https://arxiv:cs.LG/1811.09904.
[145] 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*. Springer International Publishing, Cham, 92–104.
[146] Shihao Shen, Yiwen Han, Xiaofei Wang, and Yan Wang. 2019. Computation offloading with multiple agents in edge-computing–supported IoT. *ACM TOSN* 16, 1 (2019), Article 8.
[147] Wenqi Shi, Sheng Zhou, and Zhisheng Niu. 2019. Device scheduling with fast convergence for wireless federated learning. Retrieved from https://arXiv:1911.00856.
[148] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi. 2019. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In *Proceedings of IEEE ISBI*. 270–274.
[149] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. In *Proceedings of NIPS'17*. Curran Associates, 4427–4437.
[150] Chunhe Song, Tong Li, Xu Huang, Zhongfeng Wang, and Peng Zeng. 2019. *Towards Edge Computing Based Distributed Data Analytics Framework in Smart Grids*. Springer International Publishing, Cham, 283–292.
[151] K. Sozinov, V. Vlassov, and S. Girdzijauskas. 2018. Human activity recognition using federated learning. In *Proceedings of IEEE ISPA/IUCC/BDCloud/SocialCom/SustainCom*. 1103–1111.
[152] Xudong Sun, Andrea Bommert, Florian Pfisterer, Jörg Rähenfürher, Michel Lang, and Bernd Bischl. 2020. High dimensional restrictive federated model selection with multi-objective Bayesian optimization over shifted distributions. In *Proceedings of IntelliSys*, Yaxin Bi, Rahul Bhatia, and Supriya Kapoor (Eds.). Cham, 629–647.
[153] Yuxuan Sun, Sheng Zhou, and Deniz Gündüz. 2019. Energy-aware analog aggregation for federated learning with redundant data. Retrieved from https://arXiv:1911.00188.
[154] Manoj A Thomas, Diya Suzanne Abraham, and Dapeng Liu. 2018. Federated machine learning for translational research. In *Proceedings of AMCIS'18*.
[155] Aleksei Triastcyn and Boi Faltings. 2019. Federated Generative Privacy. Retrieved from https://arxiv:stat.ML/1910.08385.
[156] Aleksei Triastcyn and Boi Faltings. 2019. Federated learning with Bayesian differential privacy. Retrieved from https://arXiv:1911.10071.
[157] M. Troglia, J. Melcher, Y. Zheng, D. Anthony, A. Yang, and T. Yang. 2019. FaIR: Federated incumbent detection in CBRS band. In *Proceedings of IEEE DySPAN*. 1–6.
[158] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of AISec'19*. ACM, 1–11.
[159] Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. 2019. *Functional Federated Learning in Erlang (ffl-erl)*. Springer International Publishing, Cham, 162–178.
[160] D. Verma, G. White, and G. de Mel. 2019. Federated AI for the enterprise: A web services based implementation. In *Proceedings of IEEE ICWS*. 20–27.
[161] Dinesh C. Verma, Graham White, Simon Julier, Stepehen Pasteris, Supriyo Chakraborty, and Greg Cirincione. 2019. Approaches to address the data skew problem in federated learning. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, Vol. 11006. International Society for Optics and Photonics, 110061I.
[162] Tung T. Vu, Duy T. Ngo, Nguyen H. Tran, Hien Quoc Ngo, Minh N. Dao, and Richard H. Middleton. 2019. Cell-free massive MIMO for wireless federated learning. Retrieved from https://arXiv:1909.12567.
[163] Z. Wan, X. Xia, D. Lo, and G. C. Murphy. 2019. How does machine learning change software development practices? *IEEE Trans. Softw. Eng.* (2019).
[164] Guan Wang. 2019. Interpret federated learning with shapley values. Retrieved from https://arXiv:1905.04519.
[165] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. 2019. Measure contribution of participants in federated learning. Retrieved from https://arXiv:1909.08525.
[166] Kangkang Wang, Rajiv Mathews, Chloe Kiddon, Hubert Eichner, Francoise Beaufays, and Daniel Ramage. 2019. Federated Evaluation of On-device Personalization. Retrieved from https://arxiv:cs.LG/1910.10252.
[167] L. Wang, W. Wang, and B. Li. 2019. CMFL: Mitigating communication overhead for federated learning. In *Proceedings of IEEE ICDCS*. 954–964.
[168] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE J.-SAC* 37, 6 (June 2019), 1205–1221.
[169] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farokhi Farhad, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2019. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. Retrieved from https://arxiv:cs.LG/1911.00222.

[170] Xiguang Wei, Quan Li, Yang Liu, Han Yu, Tianjian Chen, and Qiang Yang. 2019. Multi-agent visualization for explaining federated learning. In *Proceedings of IJCAI*. AAAI Press, 6572–6574.
[171] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo. 2019. DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Trans. Dependable Secure Comput.* (2019).
[172] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. 2005. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.* 11, 1 (Dec. 2005), 102–107.
[173] Wentai Wu, Ligang He, Weiwei Lin, Stephen Jarvis, et al. 2019. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. Retrieved from https://arXiv:1910.01355.
[174] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Asynchronous Federated Optimization. Retrieved from https://arxiv:cs.DC/1903.03934.
[175] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin. 2020. VerifyNet: Secure and verifiable federated learning. *IEEE Trans. Info. Forensics Secur.* 15 (2020), 911–926.
[176] Jie Xu and Fei Wang. 2019. Federated Learning for Healthcare Informatics. Retrieved from https://arxiv:cs.LG/1911.06270.
[177] Mengwei Xu, Feng Qian, Qiaozhu Mei, Kang Huang, and Xuanzhe Liu. 2018. DeepType: On-device deep learning for input personalization service with minimal privacy concern. *Proc. IMWUT* 2, 4 (2018), Article 197.
[178] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. 2019. HybridAlpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of AISec'19*. ACM, 13–23.
[179] Zichen Xu, Li Li, and Wenting Zou. 2019. Exploring federated learning on battery-powered devices. In *Proceedings of TURC'19*. ACM, Article 6.
[180] Howard H Yang, Ahmed Arafa, Tony QS Quek, and H Vincent Poor. 2019. Age-based scheduling policy for federated learning in mobile edge networks. Retrieved from https://arXiv:1910.14648.
[181] Kai Yang, Tao Fan, Tianjian Chen, Yuanming Shi, and Qiang Yang. 2019. A quasi-Newton method based vertical federated learning framework for logistic regression. Retrieved from https://arXiv:1912.00513.
[182] K. Yang, T. Jiang, Y. Shi, and Z. Ding. 2020. Federated learning via over-the-air computation. *IEEE Trans. Wireless Commun.* 19, 3 (2020), 2022–2035.
[183] 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.
[184] Shengwen Yang, Bing Ren, Xuhui Zhou, and Liping Liu. 2019. Parallel distributed logistic regression for vertical federated learning without third-party coordinator. Retrieved from https://arXiv:1911.09824.
[185] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving Google keyboard query suggestions. Retrieved from https://arXiv:1812.02903.
[186] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. 2019. *FFD: A Federated Learning Based Method for Credit Card Fraud Detection.* Springer International Publishing, Cham, 18–32.
[187] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. 2019. Energy efficient federated learning over wireless communication networks. Retrieved from https://arXiv:1911.02417.
[188] X. Yao, C. Huang, and L. Sun. 2018. Two-stream federated learning: Reduce the communication costs. In *Proceedings of IEEE VCIP*. 1–4.
[189] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun. 2019. Towards faster and better federated learning: A feature fusion approach. In *Proceedings of IEEE ICIP*. 175–179.
[190] D. Ye, R. Yu, M. Pan, and Z. Han. 2020. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access* 8 (2020), 23920–23935.
[191] B. Yin, H. Yin, Y. Wu, and Z. Jiang. 2020. FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things. *IEEE IoT-J.* 7, 7 (2020), 6348–6359.
[192] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas. 2018. Federated learning based proactive content caching in edge computing. In *Proceedings of IEEE GLOBECOM*. 1–6.
[193] Song Guo Yufeng Zhan, Peng Li. 2020. Experience-driven computational resource allocation of federated learning by deep reinforcement learning. In *Proceedings of IEEE IPDPS*.
[194] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. Retrieved from https://arXiv:1905.12022.
[195] Qunsong Zeng, Yuqing Du, Kin K. Leung, and Kaibin Huang. 2019. Energy-efficient radio resource allocation for federated edge learning. Retrieved from https://arxiv:1907.06040.
[196] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo. 2020. A learning-based incentive mechanism for federated learning. *IEEE IoT-J.* 7, 7 (2020), 6360–6368.

[197] Jiale Zhang, Junyu Wang, Yanchao Zhao, and Bing Chen. 2019. *An Efficient Federated Learning Scheme with Differential Privacy in Mobile Edge Computing*. Springer International Publishing, Cham, 538–550.

[198] X. Zhang, X. Chen, J. Liu, and Y. Xiang. 2019. DeepPAR and DeepDPA: Privacy-preserving and asynchronous deep learning for industrial IoT. *IEEE Trans. Industr. Inform.* 16, 3 (2019), 2081–2090.

[199] X. Zhang, M. Peng, S. Yan, and Y. Sun. 2019. Deep reinforcement learning based mode selection and resource allocation for cellular V2X communications. *IEEE IoT-J.* 7, 7 (2019), 6380–6391.

[200] Ying Zhao, Junjun Chen, Di Wu, Jian Teng, and Shui Yu. 2019. Multi-task network anomaly detection using federated learning. In *Proceedings of SoICT'19*. ACM, 273–279.

[201] Ying Zhao, Junjun Chen, Jiale Zhang, Di Wu, Jian Teng, and Shui Yu. 2020. PDGAN: A novel poisoning defense method in federated learning using generative adversarial network. In *Proceedings of ICA3PP*, Sheng Wen, Albert Zomaya, and Laurence T. Yang (Eds.). Springer International Publishing, Cham, 595–609.

[202] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. Retrieved from https://arXiv:1806.00582.

[203] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, and Dusit Niyato. 2019. Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system. Retrieved from https://arXiv:1906.10893.

[204] P. Zhou, K. Wang, L. Guo, S. Gong, and B. Zheng. 2019. A privacy-preserving distributed contextual federated online learning framework with big data support in social recommender systems. *IEEE Trans. Knowl. Data. Eng.* 33, 3 (2019), 824–838.

[205] W. Zhou, Y. Li, S. Chen, and B. Ding. 2018. Real-time data processing architecture for multi-robots based on differential federated learning. In *Proceedings of IEEE SmartWorld*. 462–471.

[206] G. Zhu, Y. Wang, and K. Huang. 2020. Broadband analog aggregation for low-latency federated edge learning. *IEEE Trans. Wirel. Commun.* 19, 1 (Jan. 2020), 491–506.

[207] H. Zhu and Y. Jin. 2020. Multi-objective evolutionary federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 4 (2020), 1310–1322.

[208] Xudong Zhu, Hui Li, and Yang Yu. 2019. *Blockchain-based Privacy Preserving Deep Learning*. Springer International Publishing, Cham, 370–383.

[209] 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 *Proceedings of IEEE SmartData*. 874–879.

[210] W. Zhang et al. 2021. Blockchain-based federated learning for device failure detection in industrial IoT. *IEEE Internet of Things Journal* 8, 7 (2021), 5926–5937.

[211] W. Zhang et al. 2021. Dynamic fusion-based federated learning for COVID-19 Detection. *IEEE Internet of Things Journal*.

Received July 2020; revised December 2020; accepted February 2021