Federated Learning: Opportunities and Challenges

Priyanka Mary Mammen
University of Massachusetts, Amherst
pmammen@cs.umass.edu

ABSTRACT

Federated Learning (FL) is a concept first introduced by Google in 2016, in which multiple devices collaboratively learn a machine learning model without sharing their private data under the supervision of a central server. This offers ample opportunities in critical domains such as healthcare, finance etc, where it is risky to share private user information to other organisations or devices. While FL appears to be a promising Machine Learning (ML) technique to keep the local data private, it is also vulnerable to attacks like other ML models. Given the growing interest in the FL domain, this report discusses the opportunities and challenges in federated learning.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

Federated Learning, Distributed Systems

1 INTRODUCTION

Artificial Intelligence (AI)/Machine Learning (ML) started getting popular in the last few 4-5 years when AI beat humans in a board game named Alpha-Go[28]. Availability of Big-data and powerful computing units further accelerated the adoption of Machine Learning technologies in domains such as finance, healthcare, transportation, customer services, e-commerce, smart home applications etc. With this widespread adoption of ML techniques, it is therefore important to ensure the security and privacy of the techniques. In most of the machine learning applications, data from various organizations or devices are aggregated in a central server or a cloud platform for training the model. This is a key limitation especially when the training data set contains sensitive information and therefore, poses security threats. For example, to develop a breast cancer detection model from MRI scans, different hospitals can share their data to develop a collaborated ML model. Whereas, sharing private

patient information to a central server can reveal sensitive information to the public with several repercussions. In such scenarios, Federated Learning can be better option.Federated Learning is a collaborative learning technique among devices/organizations, where the model parameters from local models are shared and aggregated instead of sharing their local data.

The notion of Federated Learning is introduced by Google in 2016, where they first applied in google keyboard to collaboratively learn from several android phones [24]. Given that FL can be applied to any edge device, it has the potential to revolutionize critical domains such as healthcare, transportation, finance, smart home etc. The most prominent example is when the researchers and medical practitioners from different parts of the world collaboratively developed an AI pandemic engine for COVID-19 diagnosis from chest scans [1]. Another interesting application would be in transportation networks, for training the vehicles for autonomous driving and city route planning. Similarly for smart-home applications, edge devices in different homes can collaboratively learn on context aware policies using a federated learning framework [38].

While the applications are many, there are several challenges associated with federated learning. The challenges can be broadly classified into two: training-related challenges and security challenges. Training related challenges encompass the communication overhead during multiple training iterations, heterogeneity of the devices participating in the learning and heterogeneity of data used for training .Whereas security challenges include the privacy and security threats due to the presence of adversaries ranging from malicious clients in the local device to a malicious user who has only a black-box access to the model. In FL, although the private data does not leave the device, it might be still possible for an adversary or a curious observer to learn the presence of a data point used for training in the local models. In order to overcome this attack, some kind of cryptographic technique is required to keep the information differentially private [10]. Whereas security attacks can be mostly induced by the presence of malicious clients in the learning, and they can be either targeted or non-targeted. In targeted attacks, the adversary wants to manipulate the labels on specific tasks. Whereas in non-targeted attacks, the motivation of the adversary is just to compromise the accuracy of the global model. The defense mechanisms require to detect malicious devices and remove them from further learning or nullify the effect on the global model induced by the malicious devices [8].

Numerous research efforts have been undertaken in the last few years to fortify the Federated Learning domain and it has some effect on the performance parameters like accuracy, computational costs etc. Being a distributed system,its difficult to identify malicious participants in FL. FL domain has grown so far from being a small application introduced by Google. Researchers have introduced various FL architectures, incentive mechanisms to foster user participation, cloud services for FL etc. Motivated by this growing
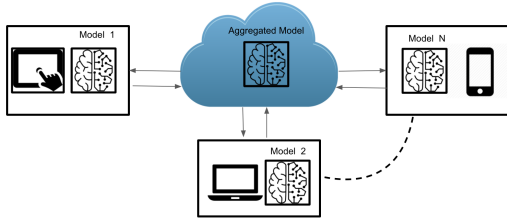
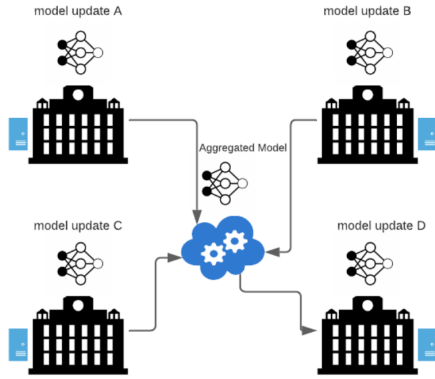Figure 1: Overview of Federated Learning across devices.

Figure 2: Overview of Federated Learning across organisations

interest in the Federated Learning domain, we present this survey paper. The recent works [2, 14, 26, 36] are focused either on different federated learning architecture or on different challenges in FL domain. Whereas so many interesting advancements are taking place beyond applying it to new application domains and fostering the security aspects Therefore, in this survey we also try to cover the recent developments along with providing give a general overview on Federated Learning applications and security concerns in the domain.

2 OVERVIEW OF FEDERATED LEARNING

A simple representation of Federated Learning is shown in fig. 1 and fig. 2. Federated Learning has primarily four main steps:

- Client Selection/Sampling : Server either randomly picks desired participants from a pool of devices or use some algorithm for client selection. [16, 37, 39] talk about some client selection techniques for FL.
- Parameter Broadcasting: Server broadcasts the global model parameters to the selected clients
- Local Model Training: The clients will parallelly retrain the models using their local data.
- Model Aggregation: Clients will send back their local model parameters to the server and model parameters will be aggregated towards the global model.

The above steps will be repeated in an iterative manner for n times as desired.

3 TYPES OF FEDERATED LEARNING

In this section, we introduce different types of Federated Learning frameworks.

- Vertical Federated Learning -Vertical Federated Learning is used for cases in which each device contains dataset with different features but from sample instances. For instance, two organisations have data about the same group of people with different feature set can use Vertical FL to build a shared ML model.
- Horizontal Federated Learning - Horizontal Federated Learning is used for cases in which each device contains dataset with the same feature space but with different sample instances. The first use case of FL- Google keyboard uses this type of learning in which the participating mobile phones have different training data with same features.
- Federated Transfer Learning - Federated Transfer learning is similar to the traditional Machine Learning, where we want to add a new feature on a pre-trained model. The best example would be for giving an extension to the vertical federated learning - If we want to extend the ML to more number of sample instances which are not present in all of the collaborating organisations.
- Cross-Silo Federated Learning - Cross- Silo Federated Learning is used when the participating devices are less in number and available for all rounds. The training data can be in horizontal or vertical FL format. Mostly cross-silo is used for cases with organisations. Works such as [40] use cross-silo FL to develop their model.
- Cross-Device Federated Learning - Scenarios with a large number of participating devices use Cross-device Federated Learning. Client-selection and incentive designs[37] are some notable techniques needed to facilitate this type of FL.

4 APPLICATIONS

4.1 Healthcare

Electronic Health Records (EHR) is considered as the main source of healthcare data for machine learning applications [11]. If ML models are trained only using the limited data available in a single hospital, it might introduce some amount of bias in the predictions. Thus, to make the models more generalizable, it requires training with more data, which can be realized by sharing data among organizations. Given the sensitive nature of the healthcare data, it might not be feasible to share the electronic health records of patients among hospitals. In such situations, federated learning can serve as an option for building a collaborative learning model for healthcare data.

4.2 Transportation

With the increase in the ubiquity of sensors in vehicular networks, it is feasible to capture more data and train ML models. Machine

Learning based models are generally applied to both vehicle management and traffic management [32]. The current autonomous driving decisions are limited by the dynamic nature of the surroundings as the training is carried out offline. FL can rescue such situations by online training vehicles from different geographical locations which can facilitate accurate labelling of the features. Similarly for traffic flow prediction techniques, a large amount of data is required, but most of the data is divided among various organizations and cannot be exchanged to protect the privacy [21]. To address such situations also, we can deploy FL methods.

4.3 Finance

One best use of federated learning in finance is in the banking sector, for loan risk assessment [6]. Normally banks use whitelisting techniques to rule out the customers using their credit card reports from the central banks. Factors such as taxation, reputation etc can also be utilized for risk management by collaborating with other finance institutions and e-commerce companies. As it is risky to share private information of customers among organizations, they can make use of FL to build a risk assessment ML model.

4.4 Natural Language Processing

Natural Language Processing (NLP) is one of the most common applications which is built on machine learning models. It helps us understand human language semantics in a better way. However, it requires huge amount of data to train highly accurate language models. This data can be easily gathered from mobile phones, tablets etc. Again, privacy comes as a bottleneck here for centralized language learning models, as the textual information from each edge device contains user information. In [9], the authors have shown that it is feasible to build NLP models using a FL framework.

5 TRAINING BOTTLENECKS

Being a distributed system, Federated Learning faces several challenges during the training time.

5.1 Communication Overheads

Communication overheads is one of the major bottlenecks in federated learning. Existing works try to solve this by either data compression [17] or by allowing only the relevant outputs by the clients[13, 22] to be sent back to the central server.

5.2 Systems and Data Heterogeneity

The heterogeneity of the systems in the network as well as the non-identically distributed data from the devices affect the performance of the FL model [20, 23]. Although, FedAvg is introduced as a method to tackle the heterogeneity, it is still not robust enough to systems heterogeneity. Works such as [5, 21] try to address this problem by modifying the model aggregation methods.

6 PRIVACY AND SECURITY CONCERNS

Like any machine learning model, Federated Learning models are also prone to attacks. The attacks can be introduced by a compromised central server or compromised local devices in the learning framework or by any participant in the FL workflow. Attacks in the context of FL will be discussed in this section.

6.1 Membership Inference Attacks

Although the raw user data does not leave the local device, there are still many ways to infer the training data used in FL. For instance, in some scenarios, it is possible to infer the information about the training data from the model updates during the learning process. The defense measure looks for mechanism offering a differential privacy guarantee. The most common techniques are secure computation, differential privacy schemes and running in a trusted execution environment.

6.1.1 Defense Mechanism 1: Secure Computation. Two main techniques come under Secure Computation: Secure Multiparty Computation (SMC) and Homomorphic Encryption. In SMC, two or more parties agree to perform the inputs provided by the participants and reveal the outputs only to a subset of participants. Whereas in homomorphic encryption, computations are performed on encrypted inputs without decrypting first.

6.1.2 Defense Mechanism 2: Differential Privacy (DP). In differential privacy schemes, the contribution of a user is masked by adding noise to the clipped model parameters before model aggregation [10]. Some amount of model accuracy will be lost by adding noise to the parameters.

6.1.3 Defense Mechanism 3:Trusted Execution Environment. Trusted Execution Environment(TEE) provides a secure platform to run the federated learning process with low computational overhead when compared to secure computation techniques [25]. The current TEE environment is suitable only to CPU devices.

6.2 Data Poisoning Attacks

Data Poisoning Attacks are the most common attacks against ML models. In order to launch a data poisoning attack in FL model, adversary poison the training data in a certain number of devices participating in the learning process so that the global model accuracy is compromised. The adversary can poison the data either by directly injecting poisoned data to the targeted device or injecting poisoned data through other devices [30]. Such attacks can be either targeted or non-targeted. By targeted it means, the adversaries want to influence on the prediction of a subset of classes while deteriorating the global model accuracy[33].

6.2.1 Defense Mechanisms. The defense mechanism against such attacks is to identify the malicious participants based on their model updates before model averaging in each round of learning.

6.3 Model Poisoning Attacks

Model Poisoning attacks are similar to data poisoning attacks, where the adversary tries to poison the local models instead of the local data. The major motivation behind the model poisoning attack is to introduce errors in the global model. The adversary launches a model poisoning attack by compromising some of the devices and modifying it's local model parameters so that the accuracy of the global model is affected.

6.3.1 Defense Mechanisms. The defenses against model poisoning attacks are similar to data poisoning attacks. The most common defense measures are rejections based on Error rate and Loss function

[8]. In error rate based rejections, the models with significant impact on the error rate on the global model will be rejected. Whereas in loss function based rejections, the models will be rejected based on their impact on loss function of the global model. In some cases, rejections can be made by combining both error based rejections and loss function based rejections.

6.4 Backdoor Attacks

Secure averaging in federated learning lets the devices to be anonymous during the model updating process. Using the same functionality, a device or a group of devices can introduce a backdoor functionality in the global model of federated learning [3]. Using a backdoor, an adversary can mislabel certain tasks without affecting the accuracy of the global model. For instance, an attacker can choose a specific label for a data instance with specific characteristics. Backdoor attacks are also known as targeted attacks. The Intensity of such attacks depends on the proportion of the compromised devices present and model capacity of federated learning [31] .

6.4.1 Defense Mechanisms. : The defense against backdoor attacks is either weak differential privacy or norm thresholding of updates. Participant level differential privacy can serve as a form of defense against such attacks but at the cost of performance of the global model [10]. Whereas norm thresholding can be applied to remove models with boosted model parameters. Even with this defense measures, it is hard to find the malicious participants owing to the secure aggregation techniques and capacity of the deep learning model. Also FL framework being a distributed system, it might be even harder to manage the randomly misbehaving devices.

7 RECENT DEVELOPMENTS IN FL

7.1 One-shot federated Learning

In most of the federated learning frameworks, there will be multiple rounds of communication between devices and the central server, which increases the communication overheads. Recently, there is a growing interest in one-shot federated learning which is first introduced by [12], where the global model is learned in a single round of communication. In order to overcome communication overheads of sending bulky gradients, [41] proposes a distilled one-shot federated learning, where each device distills their data and send the fabricated data to the central server. The server then learns the global model by training over the combined data from all the devices.

7.2 Incentive Mechanisms

Current FL approaches work under the assumption that devices will cooperate in the learning process whenever required without considering the rewards. Whereas in actual practice, devices or clients must be economically compensated for their participation. To encourage/improve device participation in FL, works such as [15, 16, 37, 39] propose a reputation based incentive mechanism i.e, devices get rewards based on their model accuracy, data reliability and contribution to the global model. However, these works did not talk about how to model convergence and additional communication overheads induced into the framework.
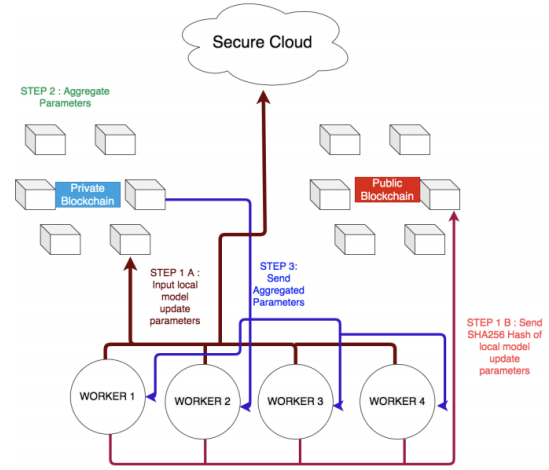

Figure 3: A sample architecture for federated learning over blockchain.

7.3 Federated Learning as a Service

Machine Learning as a Service is getting popular these days and most of them offer only centralized services. In order to offer Federated Learning as a cloud service, it should consider collaboration among third party applications. A recent work [18] tried to develop a FL framework (as a service) which allows applications of third parties to contribute and collaborate on a ML model. The framework is claimed to be suitable for any operation environment as well.

7.4 Asynchronous Federated Learning

Most of the current FL aggregation techniques are designed for devices working in a synchronized manner. However, due to systems and data heterogeneity, training and model transfer occur in a asynchronized manner. Therefore it might not be feasible to scale federated optimization in an synchronized manner [35]. Works such as [29, 34, 35] talks about carrying out federated learning in an asynchronous environment. Compared to FedAvg (working in a synchronized manner), asynchronous Federated averaging techniques can handle more devices and allows updates to come at a time.

7.5 Blockchain in FL

An aggregator is necessary to update the global model managing the asynchronous arrival of parameters from the devices. This can be a constraint for the widespread adoption for the FL models. As blockchain is a decentralized network, devices can collaboratively learn without the central aggregator. Works such as [4, 7, 19, 27] propose Federated Learning in a block-chain framework. A sample architecture [7] is shown in fig 3.

8 FINAL REMARKS

Federated Learning offers a secure collaborative machine learning framework for different devices without sharing their private data.

This attracted a lot of researchers and there is extensive research happening in this domain. Federated Learning has been applied in several domains such as healthcare, transportation etc. Although FL frameworks offer a better privacy guarantee than other ML frameworks, it is still prone to several attacks. The distributed nature of the framework makes it even harder to deploy defense measures. For instance,the gaussian noise added to the local models (for DP) can confuse the aggregation schemes and may result in leaving out the benign participants (while applying model poisoning defense measures). So therefore an interesting research question to pursue will be: Is it possible to develop a byzantine tolerant FL model while ensuring user privacy using schemes with low computational cost?

ACKNOWLEDGMENTS

REFERENCES

[1] 2020. *AI pandemic engine.* https://hai.stanford.edu/blog/pandemic-ai-engine-without-borders.
[2] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. 2020. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* 8 (2020), 140699–140725.
[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics.* PMLR, 2938–2948.
[4] Xianglin Bao, Cheng Su, Yan Xiong, Wenchao Huang, and Yifei Hu. 2019. Flchain: A blockchain for auditable federated learning with trust and incentive. In *2019 5th International Conference on Big Data Computing and Communications (BIGCOM).* IEEE, 151–159.
[5] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046* (2019).
[6] Yong Cheng, Yang Liu, Tianjian Chen, and Qiang Yang. 2020. Federated learning for privacy-preserving AI. *Commun. ACM* 63, 12 (2020), 33–36.
[7] Harsh Bimal Desai, Mustafa Safa Ozdayi, and Murat Kantarcioglu. 2020. BlockFLA: Accountable Federated Learning via Hybrid Blockchain Architecture. *arXiv preprint arXiv:2010.07427* (2020).
[8] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to Byzantine-robust federated learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20).* 1605–1622.
[9] Daniel Garcia Bernal. 2020. Decentralizing Large-Scale Natural Language Processing with Federated Learning.
[10] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).
[11] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L Beam, Irene Y Chen, and Rajesh Ranganath. 2020. A Review of Challenges and Opportunities in Machine Learning for Health. *AMIA Summits on Translational Science Proceedings* 2020 (2020), 191.
[12] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-shot federated learning. *arXiv preprint arXiv:1902.11175* (2019).
[13] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R Ganger, Phillip B Gibbons, and Onur Mutlu. 2017. Gaia: Geo-distributed machine learning approaching {LAN} speeds. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17).* 629–647.
[14] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).
[15] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. 2019. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS).* IEEE, 1–5.
[16] Latif U Khan, Shashi Raj Pandey, Nguyen H Tran, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. 2020. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine* 58, 10 (2020), 88–93.
[17] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
[18] Nicolas Kourtellis, Kleomenis Katevas, and Diego Perino. 2020. FLaaS: Federated Learning as a Service. *arXiv preprint arXiv:2011.09359* (2020).
[19] Rajesh Kumar, Abdullah Aman Khan, Sinmin Zhang, WenYong Wang, Yousif Abuidris, Waqas Amin, and Jay Kumar. 2020. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. *arXiv preprint arXiv:2007.06537* (2020).
[20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.
[21] Yi Liu, JQ James, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. 2020. Privacy-preserving Traffic Flow Prediction: A Federated Learning Approach. *IEEE Internet of Things Journal* (2020).
[22] WANG Luping, WANG Wei, and LI Bo. 2019. Cmfl: Mitigating communication overhead for federated learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS).* IEEE, 954–964.
[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics.* PMLR, 1273–1282.
[24] Brendan McMahan and Daniel Ramage. 2017. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog* 3 (2017).
[25] Fan Mo and Hamed Haddadi. 2019. Efficient and private federated learning using tee. In *EuroSys.*
[26] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. 2020. A survey on security and privacy of federated learning. *Future Generation Computer Systems* 115 (2020), 619–640.
[27] Paritosh Ramanan, Kiyoshi Nakayama, and Ratnesh Sharma. 2019. BAFFLE: Blockchain based aggregator free federated learning. *arXiv preprint arXiv:1909.07452* (2019).
[28] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.
[29] Michael R Sprague, Amir Jalalirad, Marco Scavuzzo, Catalin Capota, Moritz Neun, Lyman Do, and Michael Kopp. 2018. Asynchronous federated learning for geospatial applications. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 21–28.
[30] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, and Ji Liu. 2020. Data Poisoning Attacks on Federated Machine Learning. *arXiv preprint arXiv:2004.10020* (2020).
[31] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. 2019. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963* (2019).
[32] Kang Tan, Duncan Bremner, Julien Le Kernec, and Muhammad Imran. 2020. Federated Machine Learning in Vehicular Networks: A summary of Recent Applications. In *2020 International Conference on UK-China Emerging Technologies (UCET).* IEEE, 1–4.
[33] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data Poisoning Attacks Against Federated Learning Systems. In *European Symposium on Research in Computer Security.* Springer, 480–501.
[34] Marten van Dijk, Nhuong V Nguyen, Toan N Nguyen, Lam M Nguyen, Quoc Tran-Dinh, and Phuong Ha Nguyen. 2020. Asynchronous Federated Learning with Reduced Number of Rounds and with Differential Privacy from Less Aggregated Gaussian Noise. *arXiv preprint arXiv:2007.09208* (2020).
[35] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934* (2019).
[36] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.
[37] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A Sustainable Incentive Scheme for Federated Learning. *IEEE Intelligent Systems* (2020).
[38] Tianlong Yu, Tian Li, Yuqiong Sun, Susanta Nanda, Virginia Smith, Vyas Sekar, and Srinivasan Seshan. 2020. Learning Context-Aware Policies from Multiple Smart Homes via Federated Multi-Task Learning. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI).* IEEE, 104–115.
[39] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. 2020. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal* (2020).
[40] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *2020 {USENIX} Annual Technical Conference ({USENIX} {ATC} 20).* 493–506.
[41] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. Distilled One-Shot Federated Learning. *arXiv preprint arXiv:2009.07999* (2020).