# Fast Meta Failure Recovery for Federated Meta-Learning [Vision Paper]

Hailu Xu, Brandon Delliquadri, Chao Wang[§], Zhengxiong Li[‡], Hui Luo[†]
California State University, Long Beach
University of Colorado Denver[‡], CARS[§], University of Wollongong[†]
Email:{hailu.xu@csulb.edu, chaowangasaph@gmail.com, zhengxiong.li@ucdenver.edu, huil@uow.edu.au}

***Abstract*—In recent years, the field of distributed deep learning within the Internet of Things (IoT) or the edge has experienced exponential growth. Federated meta-learning has emerged as a significant advancement, enabling collaborative learning among source nodes to establish a global model initialization. This approach allows for optimal performance while necessitating minimal data samples for updating model parameters at the target node. Federated meta-learning has gained increased attention due to its capacity to provide real-time edge intelligence. However, a critical aspect that remains inadequately explored is the recovery of interim meta knowledge's failure, which constitutes a pivotal key for adapting to new tasks. In this paper, we introduce FMRec, a novel platform designed to offer a fast and flexible recovery mechanism for failed interim meta knowledge in various federated meta-learning scenarios. FMRec serves as a complementary system compatible with different types of federated models and is adaptable to diverse tasks. We present a demonstration of its design and assess its efficiency and reliability through real-world applications.**

***Index Terms*—Federated Meta-Learning; Failure Recovery; IoT.**

## I. Introduction

Federated meta-learning (FML) is an advanced machine learning approach that combines federated learning and meta-learning [13]. Federated meta-learning allows source nodes collaboratively learn a global model initialization, so that maximal performance can be obtained with the model parameters updated with only a few data samples at the target node [3]. In FML, a central model is initially trained using meta-learning principles to quickly adapt to new tasks. This model is then distributed to a network of decentralized devices or servers, each with its own dataset and tasks. These devices/servers fine-tune the model using their local data, allowing them to specialize for their specific tasks while retaining the ability to adapt to new tasks efficiently. Aggregating the updated models from these devices refines the central model, ensuring that the collective knowledge gained from various devices enhances the model's generalization capabilities [12].

Federated meta-learning are applicable to numerous domains such as healthcare, finance, and Internet of Things (IoT) [5, 6, 14]. For instance, in healthcare areas, medical institutions can collaborate without sharing sensitive patient data. Instead, they share their model updates, which benefits from the diverse patient populations encountered across different hospitals. Models in FML become increasingly adept at rapid adaptation to evolving tasks, thereby achieving real-time intelligence in IoT/edge devices or servers.

The "meta" in federated meta-learning addresses the challenge of adapting a model across different tasks within a federated network. The underlying rationale behind the meta knowledge is to train the model's initial parameters over many tasks, such that the pre-trained model can achieve maximal performance on a new task after quick adaptation using only a small amount of data corresponding to that new task. In FML, this meta-learning component ensures that the model becomes proficient not only in adapting to tasks within individual devices but also in generalizing its knowledge to tasks across the entire federated network.

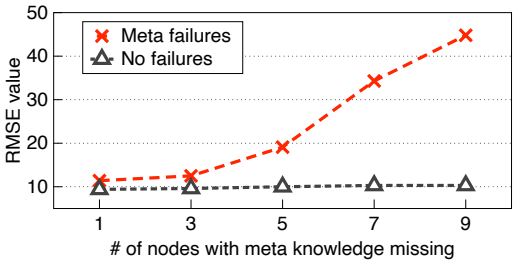


Fig. 1. The performance degradation with different number of nodes losing their meta knowledge (RMSE is the root mean square error).

However, FML encounters significant challenges associated with the potential failure of meta knowledge.

Meta knowledge can fail due to many circumstances, such as communication errors (e.g., network errors or transmission issues), and loss of control or node unavailability (e.g., temporary network issues, device shutdown or restart) [15]. Meta knowledge failure in FML encompasses challenges pertaining to the higher-level learning process. This includes risks of catastrophic forgetting as the model fine-tunes for new tasks, potential instability in the aggregated model due to conflicting device updates, variable performance across tasks with different complexities, and optimization difficulties in fast convergence [5, 14]. Figure 1 shows the performance degradation when facing with multiple meta failures in a real-world application for small-sample parking occupancy prediction [16]. The Root Mean Square Error (RMSE) quantifies the disparity between predicted and actual model outcomes. In the "meta failures" scenario, meta data from multiple nodes is intentionally removed, and the initial model does not possess a mechanism for restoring this lost meta knowledge. Notably, we observe a pronounced decline in accuracy, with errors accumulating considerably in this situation (resulting in an RMSE as high as 45). In contrast, during independent runs with no such failures, the RMSE values consistently averaged around ten.

In this paper, we propose a robust platform that can supports

fast and flexible meta failure recovery in various federated meta-learning diagrams. By designing a platform that is easily implementable across different federated meta-learning models, we aim to facilitate the adoption of our solution in a wide variety of use cases, making it accessible to researchers and practitioners regardless of their specific model preferences.

We make the following contributions:

- We propose a novel platform that addresses the meta data failure problem. This platform is designed to handle the intricate process of managing meta data and ensures that the model's adaptability and generalization are maintained, thereby mitigating the adverse impacts of meta failure.
- Our proposed platform offers a universal solution that can seamlessly integrate with various types of federated meta-learning models. This flexibility is vital for accommodating the diverse range of models employed in real-world applications.

The rest of this paper is organized as follows: Section II discuss the latest relevant research work of federated meta-learning. Section III describes the details of our design and methodology. Section IV shows the evaluation results. We discuss the challenges, opportunities, and future work of this research problem in Section V.

## II. Related Work

Federated learning has seen widespread adoption in various domains, including healthcare, smart cities, IoT, recommendation systems, and Industry 4.0 [8, 9, 19]. Federated meta-learning, a relatively novel model, is gaining increasing attention across diverse domains such as cyberspace security [13], privacy preservation [4], and addressing multitask challenges on mobile devices [10].

The concept of a federated meta-learning framework was originally introduced in [3], where the authors creatively merged two powerful meta-learning techniques, namely MAML (Model-Agnostic Meta-Learning) and Meta-SGD (Meta Stochastic Gradient Descent) [11], within the framework of federated learning. This algorithm's core objective is to engage in collaborative meta-training by leveraging datasets from decentralized devices. In [12], a pioneering collaborative learning framework was introduced, wherein a model undergoes initial training on specific edge nodes-Following this initial training, the model swiftly adapts to become proficient in new tasks at designated target edge nodes, even when supplied with a limited quantity of data samples. This approach effectively addresses challenges associated with constrained computing resources and the inherent scarcity of local data resources at individual edge nodes. The ADMM-FedMeta method [17] strategically decomposes the initial optimization problem into multiple subproblems, enabling efficient parallel processing across both edge nodes and the platform. Additionally, NUFM [18] combines a non-uniform device selection scheme with a resource allocation strategy to jointly enhance convergence rates, minimize wall-clock time, and reduce energy costs in multi-access wireless systems.

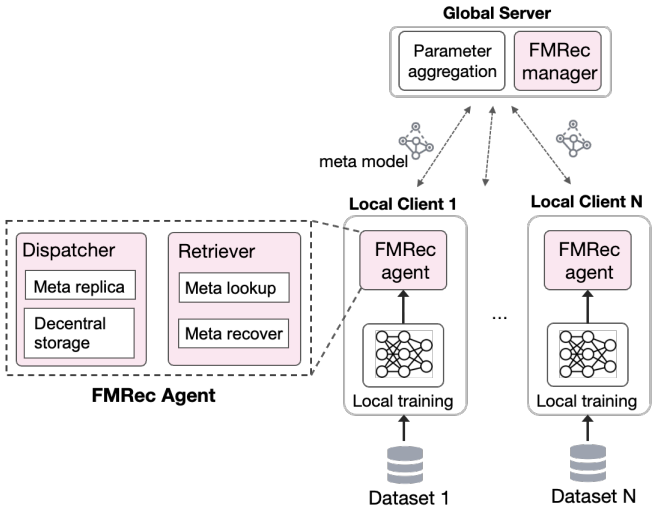


Fig. 2. The overview of FMRec.

## III. Design

In this section, we introduce the design of FMRec system, outline its workflow details, and discuss the functional components.

### A. Overview

FMRec acts as a complementary system that is built upon existing federated meta-learning models, as illustrated in Figure 2. It consists of a large set of distributed agents, which can handle the meta knowledge dispatcher and retriever for dealing with the meta failure:

- *FMRec Manager*: It orchestrates the architecture of all working agents, starts to retrieve the meta knowledge when facing with failures, and manages the distribution of local meta knowledge.
- *Dispatcher*: It is located inside each agent and responsible for the distribution of replica of meta knowledge. At runtime, it periodically handles the replica of local trained meta knowledge and saves the meta knowledge into decentralized storage for failure recovery.
- *Retriever*: It resides along with the dispatcher to fulfill the retrieve process of failure meta knowledge when FMRec manager triggers the recovery when it figures out the occurence of failures.

### B. Federated Meta-Learning Model

Followed by [3, 12], the general federated meta-learning model is defined as follows: consider a general supervised learning setting, assume tasks across edge nodes follow a meta-model, represented by a parametrized function $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^d$, where each node $i \in S$ ($S$ is the entire set of agents) has a local labeled dataset $D_i = \{(\mathbf{x}_i^1, \mathbf{y}_i^1), \ldots, (\mathbf{x}_i^j, \mathbf{y}_i^j), \ldots, (\mathbf{x}_i^{|D_i|}, \mathbf{y}_i^{|D_i|})\}$, here $|D_i|$ is the size of dataset and $(\mathbf{x}^j, \mathbf{y}^j) \in \mathcal{X} \times \mathcal{Y}$ is a sample/data point with $(\mathbf{x}_i^j, \mathbf{y}_i^j)$ follows an unknown distribution $P_i$. The experimental loss function, denoted as $L$ for the node, is

defined as $L(\boldsymbol{\theta}, D_i) \triangleq 1/|D_i| \sum_{j=1}^{j \in D_i} l(\boldsymbol{\theta}, (x_i^j, \mathbf{y}_i^j))$, where $l(\boldsymbol{\theta}, (x_i^j, \mathbf{y}_i^j))$ is the loss function. Moreover, $L_w(\boldsymbol{\theta})$ is the overall loss function across all nodes in $S$:

$$L_w(\boldsymbol{\theta}) \triangleq \sum_{i \in S} \omega_i L(\boldsymbol{\theta}, D_i),$$

where $\omega_i = |D_i| / \sum_{i \in \mathcal{S}} |D_i|$ and the weight $\omega_i$ of each edge node depends on its own local data size. In federated setting, the server maintains $\boldsymbol{\theta}$, and updates it by collecting test losses from a mini batch of nodes.

Similar to MAML [7], the designated target edge node, denoted as $t \in S$, when adapts to a new task, the model weights will be updated to $\boldsymbol{\theta}'$ and is computed by using gradient descent as

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L\left(\boldsymbol{\theta}, D_S^t\right),$$

with $\alpha$ is the learning rate, $D_S^t$ is the support set (training set) of node $t$, and then evaluates the loss $L\left(\boldsymbol{\theta}', D_Q^t\right)$ for the updated model parameter $\boldsymbol{\theta}'$ based on the query set (test set) $D_Q^t$. Therefore, the overall objective of the federated meta-learning is presented as

$$\min_{\boldsymbol{\theta}} \sum_{t \in S} \omega_i L\left(\boldsymbol{\theta}', D_Q^t\right).$$

### C. Management of FMRec Agents

FMRec employs a DHT-based hierarchical tree management system, leveraging Scribe [2] and the proximity-aware Pastry overlay [1]. This architecture efficiently organizes a substantial number of agents in real-time. To cater to diverse tasks with varying requirements and configurations, multiple trees are dynamically created within the system. Each tree facilitates application-level group communication and upholds a spanning tree comprising agents. This flexibility allows agents to seamlessly join or depart from the tree, accommodating sizes ranging from hundreds to millions. Following the principles of Scribe [2], a pseudorandom key, referred to as the treeId, is utilized to designate each tree. Typically, the treeId is generated by hashing the textual name of the tree concatenated with the name of its associated task. Agents can route JOIN messages towards the treeId, ensuring that messages reliably reach the intended agent within the tree. This approach adeptly supports a large number of agents with dynamically changing memberships, guided by the routing policy established by Scribe.

**Why tree structure**? Tasks naturally exhibit a distributed nature and considerable variability across different situations and environments. The selection of grouped agents for a given task showcases flexibility, contingent upon factors such as data volume, model training requirements, and the dynamic network environment. Consequently, the design philosophy behind FMRec is anchored in delivering adaptability and accessibility across a spectrum of runtime distributed models and tasks. FMRec adopts a functional tree structure that seamlessly aligns with the dynamic composition of FMRec agents, allowing for the straightforward adjustment of group memberships. Furthermore, the versatility of routing, guided by the treeId, facilitates the smooth transition of agents to specific task groups in response to the dynamic evolution of tasks.

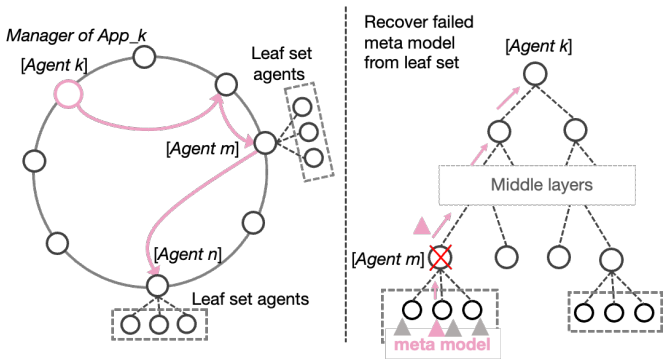


Fig. 3. The DHT-based overlay routing and meta knowledge recovery in the functional tree.

Within the functional trees, agents engage in a periodic exchange of alive messages with their upper-tier agents along the tree structure. These messages are systematically routed upwards through the tree until they ultimately reach the root node (manager). Notably, this message routing process is accomplished with a depth of $O(\log N)$, ensuring efficiency of communications. Consequently, the root agent, situated on the global server, can swiftly aggregate real-time performance data from each participating agent within the tree.

### D. Management of Meta Knowledge

One of major novel design of FMRec is the arrangement and management of meta knowledge (i.e., model parameters and test loss) at runtime. All agents are organized by a Distributed Hashtable (DHT)-based consistent ring overlay. Within this overlay, all agents assume equivalent roles, with the capacity to both provide and request services as needed. Each agent possesses a functional leaf set, comprising a predetermined number of neighboring agents whose *agentIds* are numerically closest to that agent. This arrangement serves as a dual purpose: aiding in the efficient routing of messages and facilitating the reconstruction of routing tables in the event of agent failures. Importantly, the leaf set functions as a repository for storing meta knowledge and serves as a recovery mechanism for meta knowledge that may encounter failures.

The left side of Figure 3 illustrates the functionality of dispatcher which includes the process of message routing and communication among agents within the DHT-based overlay ring. In this context, the application manager (*Agent_k*) for a specific application *k* (*App_k*), takes on the role of the root agent responsible for coordinating all agents affiliated with that particular application. During application runtime, *Agent_k* efficiently forwards all control messages to its subsequent agents in the tree via its routing table, achieving this in just $O(\log N)$ steps.

Meanwhile, worker agents such as *Agent_m* and *Agent_n* actively utilize their locally available datasets to train their re-

spective local models and compute local meta knowledge. Additionally, these worker agents perform periodic checkpoints, storing their local meta knowledge within their neighboring agents found in their designated leaf sets. This strategic use of the agent's leaf set has two reasons: (i) these neighboring agents are geographically closest to the agent, ensuring the highest bandwidth availability for efficient communication, thereby reducing network latency related to bandwidth fluctuations; (ii) the agents within the leaf set consistently belong to the local cluster or rack, which not only enhance the security of the agent's local meta knowledge but also minimize the risk of local privacy breaches and violations.

The right side of Figure 3 outlines the workflow of retriever that mainly refers to the process of meta knowledge look up and recovery. In case where an agent experiences a failure or becomes a straggler, impeding the immediate upload of its meta knowledge to the global manager (as exemplified by the scenario where *Agent_m* encounters an issue in the figure), the manager agent *Agent_k* can promptly initiate a routing operation to *Agent_m*'s designated leaf set. It can search the agents in the leaf set and figure out which agent has the replica of failed meta knowledge. Here, *Agent_k* can request the neighbor agent that possesses a replica of *Agent_m*'s meta knowledge to upload to the manager and accelerate the convergence.

In future research, our focus will delve into the deployment of meta knowledge replicas and explore various strategies to fortify data privacy, curtail data leakage risks, and fortify defenses against potential external cyber threats.

## IV. Initial Evaluation Results

Experiments are conducted on 6 Google Cloud T4 GPUs instances, each with 1 GPU and 16GB GDDR6, 12 vCPUs with 85 GB memory. We use Pastry 2.1 [1] build up to 500 virtual nodes and are configured with leafset size of 12 and transport buffer size of 6MB.

The experimental evaluation is conducted on a dataset from [16] that focuses on small-sample parking occupancy prediction. To meet the criteria for federated meta-learning, we partition the training and target tasks into four temporal segments. We employ a two-tier architecture, comprising an encoder layer (with a sequence length of 6) designed to extract time-series features and a decoder layer responsible for generating predicted values. The input is (256, 6, 1), output is (256, 1), learning rate is 0.02, max epoch is 400, and the loss function is the cross-entropy error between the predicted and true class.

We use the predictive accuracy of the model as an indicator of its performance across various data volumes, for example, Root Mean Squared Error (RMSE) that is calculated by $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2}$, where $y_i$ and $\widehat{y}_i$ represent the actual and predicted value at time $i$; and $N$ is the number of samples. Additionally, we analyze the convergence speed to gauge how quickly the model stabilizes during the training process. This evaluation helps us understand the model's scalability and efficiency in making future predictions.

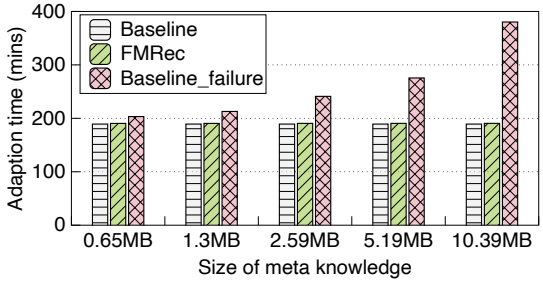


Fig. 4. The training time of adaption for the baseline model, FMRec, and baseline model with meta knowledge failures.

The evaluated model completes 400 iterations, accumulating a total metadata file size of 10.396 MB. Figure 4 illustrates the performance comparison. In the "Baseline," the optimal scenario is depicted, where meta knowledge remains constant throughout task adaptation. In Baseline_failure, nodes experiencing meta knowledge loss require restarting the meta computation. FMRec periodically snapshots the meta knowledge every 50 iterations. The results reveal that FMRec introduces only approximately one additional minute to the baseline execution time while significantly outperforming Baseline_failure, which progressively extends computation time as the file size grows. Notably, when the file size reaches 10 MB, the traditional method employed by Baseline_failure nearly doubles the execution time compared to both the baseline and FMRec. The analysis underscores FMRec's superior efficiency across observable file sizes.

## V. Challenges, Opportunities, and Future Work

**Scalability and Compatibility**: As the scale of federated networks expands with a growing number of devices or clients, the intricacy of managing model updates and aggregations escalates significantly. Furthermore, the increasing diversity of applications demanding a multitude of distributed models poses a pivotal challenge. Efficiently accommodating these heterogeneous models within a unified platform emerges as a critical research challenge within the field. The need to develop mechanisms and frameworks that can seamlessly handle a wide array of model variations, while ensuring scalability, compatibility, and effective knowledge sharing, remains a focal point in this research domain.

**Communication Overhead**: Within federated settings, the transmission of model updates between devices and a central server is a fundamental operation. In the context of meta-learning, which requires rapid adaptation based on prior meta-knowledge, this process places an added load on communication resources. In scenarios characterized by limited bandwidth or high latency, the exchange of information among a large set of nodes can become a bottleneck. Consequently, the optimization challenge lies in the delicate balance of minimizing communication overhead while upholding model accuracy, a complex endeavor that demands careful consideration and innovative solutions.

**Privacy and Security**: Federated meta-learning is often used in settings where data privacy is paramount, such as

healthcare and finance. Ensuring that sensitive data remains secure and private during the federated learning process is a challenge. Techniques like differential privacy and secure aggregation need to be incorporated. Additionally, a critical focus in our forthcoming research will be the development of innovative tools designed to effectively counter backdoor attacks and malicious control on meta knowledge, addressing a pivotal aspect of data protection in federated meta-learning environments.

**Task Complexity Variation**: In federated environments, the tasks often exhibit considerable variations in complexity. The adaptation of a single model to effectively address tasks with diverse levels of complexity poses a considerable challenge, necessitating the development of robust meta-learning strategies. In our forthcoming research endeavors, we intend to investigate the feasibility of a unified platform. Such a platform would offer a flexible and adaptable system infrastructure capable of accommodating a wide array of task complexities, thereby providing a versatile solution to the challenge of task heterogeneity in federated settings.

**Lack of Standardization:** The field of federated learning, including federated meta-learning, lacks standardized protocols and frameworks. Developing standards is crucial for fostering interoperability and facilitating meaningful comparisons among various implementations. In our current research, FMRec serves as a complementary system built upon existing federated meta-learning models. Our overarching aim is to transform this system into a standard framework capable of accommodating diverse federated meta-learning models and tasks. By doing so, we anticipate streamlining the process, reducing the need for labor-intensive platform reorganization or re-standardization efforts, and enhancing the efficiency and consistency of federated meta-learning practices.

## References


[1] Pastry. https://www.freepastry.org/FreePastry/

[2] Castro, M., Druschel, P., Kermarrec, A.M., Rowstron, A.I.: Scribe: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in communications 20(8), 1489–1499 (2002)

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

[4] Dong, F., Ge, X., Li, Q., Zhang, J., Shen, D., Liu, S., Liu, X., Li, G., Wu, F., Luo, J.: Padp-fedmeta: A personalized and adaptive differentially private federated meta learning mechanism for aiot. Journal of Systems Architecture 134, 102754 (2023)

[5] Erdol, H., Wang, X., Li, P., Thomas, J.D., Piechocki, R., Oikonomou, G., Inacio, R., Ahmad, A., Briggs, K., Kapoor, S.: Federated meta-learning for traffic steering in o-ran. In: 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall). pp. 1–7. IEEE (2022)

[6] Feng, Y., Chen, J., Xie, J., Zhang, T., Lv, H., Pan, T.: Meta-learning as a promising approach for few-shot cross-domain fault diagnosis: Algorithms, applications, and prospects. Knowledge-Based Systems (2022)

[7] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)

[8] He, J., Wang, T., Min, Y., Gu, Q.: A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. Advances in neural information processing systems (2022)

[9] Lai, Z., Oliveira, L.C., Guo, R., Xu, W., Hu, Z., Mifflin, K., Decarli, C., Cheung, S.C., Chuah, C.N., Dugger, B.N.: Brainsec: Automated brain tissue segmentation pipeline for scalable neuropathological analysis. IEEE Access 10, 49064–49079 (2022)

[10] Li, X., Li, Y., Wang, J., Chen, C., Yang, L., Zheng, Z.: Decentralized federated meta-learning framework for few-shot multitask learning. International Journal of Intelligent Systems 37(11), 8490–8522 (2022)

[11] Li, Z., Zhou, F., Chen, F., Li, H.: Meta-sgd: Learning to learn quickly for few-shot learning. arXiv preprint arXiv:1707.09835 (2017)

[12] Lin, S., Yang, G., Zhang, J.: A collaborative learning framework via federated meta-learning. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). pp. 289–299. IEEE (2020)

[13] Liu, F., Li, M., Liu, X., Xue, T., Ren, J., Zhang, C.: A review of federated meta-learning and its application in cyberspace security. Electronics 12(15), 3295 (2023)

[14] Liu, X., Deng, Y., Nallanathan, A., Bennis, M.: Federated and meta learning over non-wireless and wireless networks: A tutorial. arXiv preprint arXiv:2210.13111 (2022)

[15] Ma, X., Liao, L., Li, Z., Lai, R.X., Zhang, M.: Applying federated learning in software-defined networks: A survey. Symmetry 14(2), 195 (2022)

[16] Qu, H., Liu, S., Li, J., Zhou, Y., Liu, R.: Adaptation and learning to learn (all): An integrated approach for small-sample parking occupancy prediction. Mathematics 10(12), 2039 (2022)

[17] Yue, S., Ren, J., Xin, J., Lin, S., Zhang, J.: Inexact-admm based federated meta-learning for fast and continual edge learning. In: Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing. pp. 91–100 (2021)

[18] Yue, S., Ren, J., Xin, J., Zhang, D., Zhang, Y., Zhuang, W.: Efficient federated meta-learning over multi-access wireless networks. IEEE Journal on Selected Areas in Communications 40(5), 1556–1570 (2022)

[19] Zhan, C., Ghaderibaneh, M., Sahu, P., Gupta, H.: Deepmtl pro: Deep learning based multiple transmitter localization and power estimation. Pervasive and Mobile Computing (2022)