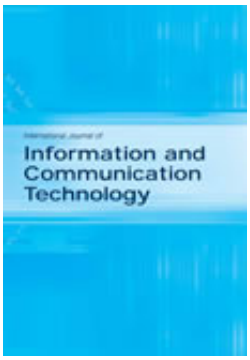
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Cross-node knowledge transfer and generalisation based on federated meta-learning in fog computing

Qiuming Zhang, Jing Luo

DOI: 10.1504/IJICT.2026.10075949

Article History:

Received:	26 October 2025
Last revised:	22 November 2025
Accepted:	25 November 2025
Published online:	06 February 2026

Cross-node knowledge transfer and generalisation based on federated meta-learning in fog computing

Qiuming Zhang

School of Computer Science,
China University of Geosciences,
Wuhan, 430074, China
Email: qmzhang@cug.edu.cn

Jing Luo*

Library and Archives,
Wuhan Vocational College of Software and Engineering,
Wuhan, 430205, China
Email: 53200121@whvcse.edu.cn
*Corresponding author

Abstract: Fog computing environments comprise numerous heterogeneous and resource-constrained nodes, resulting in significant generalisation deficiencies in conventional machine learning models when addressing cross-node tasks characterised by non-independent and identically distributed data and limited labelling resources. This research offers a federated meta-learning architecture designed for fog computing, with the objective of facilitating rapid, privacy-preserving knowledge transfer and task adaptation between nodes. This framework combines the distributed model training paradigm of federated learning with the 'learning to learn' mechanism of meta-learning. We performed comprehensive tests on public datasets to simulate standard non-IID data settings in fog computing. Results indicate that, in comparison to conventional federated learning baselines, our methodology attains an average accuracy improvement of roughly 3.7% on novel tasks, while markedly enhancing model convergence and task adaptation efficiency. This suggests that the proposed approach significantly improves learning and generalisation capabilities for resource-limited nodes in fog computing settings while addressing unfamiliar tasks.

Keywords: federated learning; meta-learning; fog computing; non-independent identically distributed; non-IID.

Reference to this paper should be made as follows: Zhang, Q. and Luo, J. (2026) 'Cross-node knowledge transfer and generalisation based on federated meta-learning in fog computing', *Int. J. Information and Communication Technology*, Vol. 27, No. 6, pp.90–107.

Biographical notes: Qiuming Zhang is an Associate Professor in the School of Computer Science at China University of Geosciences, China. He received a PhD from China University of Geosciences in 2009. His research interests include intelligent computing and machine learning.

Jing Luo is a Senior Engineer in Library and Archives at Wuhan Vocational College of Software and Engineering. She received a Master's degree from China University of Geosciences in 2009. Her research interests include intelligent education and machine learning.

1 Introduction

The internet of things and fifth-generation mobile communication technologies are moving quickly, and edge devices make a lot of data. The centralised processing model of traditional cloud computing makes it hard to deal with many problems, such as high latency, limited network bandwidth, and privacy and security concerns (Zhou et al., 2017). Fog computing has come up as a new way to do computing in this context. By putting an intermediate layer of geographically spread-out fog nodes with different resources between data sources and cloud data centres, it can provide services with low latency and a lot of location awareness. But the fact that fog computing environments are made up of many different types of devices and are spread out makes it hard to use AI models effectively (Atlam et al., 2018).

One of the biggest problems with using collaborative machine learning in fog computing is that data and tasks have certain properties that make it hard to do. Data produced by individual fog nodes often demonstrates non-independent and identically distributed (Non-IID) characteristics, indicating that data distributions among nodes might vary considerably due to differences in their geographic locations, user demographics, and functional specialties. On the other side, individual fog nodes frequently have very limited computing, storage, and energy capabilities, and hence have a hard time getting huge amounts of high-quality tagged data to train sophisticated models (Verma et al., 2022). Conventional centralised machine learning methodologies necessitate data consolidation at a central site. In fog computing, this not only goes against data privacy rules, but it also costs a lot of time and effort to communicate. Federated learning is a distributed learning method that protects privacy. It lets nodes train models locally and send only model updates, not raw data, to a server for aggregation. This is a potential way to solve these problems (Li et al., 2020).

Standard federated learning algorithms, on the other hand, mostly try to select one model that works well for all participating nodes. This model has evident problems when it comes to new activities that come up often in fog computing or nodes with very few resources. When a new fog node joins the system or an old node has to do a task that it hasn't done before, it might not have many labelled samples to learn from. In these ‘cold start’ situations, the global model trained by normal federated learning sometimes has trouble adapting fast, which leads to a big drop in generalisation performance (Zhang et al., 2021).

Federated learning, a new way of doing distributed machine learning, wants to create models together while keeping data private. Wu et al. (2020) came up with the FedSCR algorithm, which trains a global model by running several local iterations and aggregating data on a server every so often. Zhu et al. (2021) systematically established through tests that Non-IID data promotes client drift, resulting in convergence issues and performance loss in global models. Researchers have suggested a number of ways to solve this problem. For example, Huang et al. (2023) suggested using knowledge

distillation to reduce local model variance, and Zhang et al. (2024) looked into several ways to compress and send data. These studies establish the groundwork for implementing federated learning in situations with limited resources.

Meta-learning, or 'learning to learn' helps models learn from a sequence of related tasks so they can quickly adapt to new ones. Model-agnostic meta-learning (MAML) (Ng, and Guan, 2024) is a significant advancement in this domain. The main idea behind MAML is to discover a model initialisation parameter that is sensitive to the task distribution. This lets the model do well on new tasks with only a few gradient steps and a minimal amount of labelled data. This idea is quite similar to the objective of nodes in fog computing, which is to quickly adjust to changing needs. After then, Chua et al. (2021) did a lot of research and made MAML and its derivatives better, making training more stable and faster to converge. Memory-augmented neural networks (MANN), presented by Geiger et al. (2014), introduced an alternate method to meta-learning based on external memory processes, revealing a new technical avenue for addressing few-shot learning difficulties.

Combining federated learning with meta-learning has become a promising area of research in the last few years. The basic idea behind it is that each client in federated learning may be seen as a separate work, which naturally creates a meta-learning environment. Chen et al. (2023) introduced a federated meta-learning (FedFogMeta) architecture for industrial edge intelligence, adeptly tackling the issue of small-sample device defect detection while safeguarding data privacy, and markedly improving the rapid adaptability of diagnostic models to new devices. Fallah et al. (2020) presented the Per-FedAvg method through a theoretical analysis, formally combining the objective function of MAML with the optimisation process of FedAvg, so establishing a robust mathematical framework for FedFogMeta.

This research tackles this issue by merging federated learning with meta-learning to provide a FedFogMeta architecture specifically designed for fog computing. MAML is a well-known gradient-based meta-learning technique that seeks to identify optimal model startup parameters. From this point on, the model just needs one or a few gradient update steps and a minimal number of labelled data to do very well on new tasks. This study incorporates this concept into the federated learning architecture, with the objective of collectively training a globally initialised meta-model that possesses robust generalisation capabilities within a distributed setting.

The primary contributions of this paper are as follows: firstly, it formulates a formal FedFogMeta framework that systematically amalgamates distributed model training with cross-task knowledge transfer mechanisms, thereby offering a theoretical model for ongoing learning and task adaptation in fog computing environments. Second, we create and put into action a federated optimisation method that is based on MAML. This approach gradually improves the global model's starting state by combining meta-gradient updates from fog nodes at the server. This creates a strong base for quickly adapting to new jobs. Third, a lot of tests were done on a number of standard datasets to mimic real-world fog computing situations using Non-IID data. The results confirm the proposed framework's efficacy and its superiority over conventional federated learning baselines in addressing novel problems, showcasing consistent enhancements, especially in model convergence velocity and generalisation accuracy with sparse samples.

2 Related theoretical research

2.1 Federated learning

Federated learning, as a disruptive distributed machine learning paradigm, fundamentally addresses data privacy, security, and accessibility issues by training machine learning models through the collaborative effort of multiple clients (referred to as fog nodes in this context) without aggregating raw data. Its typical training workflow comprises the following core steps: first, a central server initialises a global model and distributes it to all participating clients. Subsequently, each client updates the received model using its local dataset. Finally, clients send model updates (such as gradients or weight increments) rather than raw data back to the server, which aggregates these updates to generate a new generation of the global model. This process iterates until the model converges.

Among the numerous federated learning algorithms, the FedAvg algorithm (Collins et al., 2022) stands as the most representative and widely applied classic approach. The core idea of FedAvg lies in performing a weighted average of the locally updated model parameters from clients. The update rule for its global model can be expressed by the following formula:

$$w_{t+1}=\sum_{k=1}^{K}\frac{n_k}{n}w_t^{(k)} \tag{1}$$

where w_{t+1} denotes the global model parameters after the t+1-st communication round, K is the total number of clients participating in training, n_k is the number of local data samples possessed by the k^{th} client, n is the total number of samples across all participating clients, and $w_t^{(k)}$ is the model parameters obtained by the k^{th} client after multiple rounds of local training based on the previous global model w_t. Through this approach, FedAvg effectively reduces the number of communication rounds while enabling the utilisation of distributed data with enhanced data privacy protection.

Federated learning is commonly seen as the best way to achieve collaborative intelligence in edge and fog computing environments since it is distributed and protects privacy. In fog computing, fog nodes operate as clients and use their closeness to data sources to train local models with minimal latency. After that, only updates to the model are synced with cloud servers or coordinators. This method cuts down on the amount of bandwidth used on backhaul links and lets models better adjust to local data statistics. This makes it possible to create smart solutions for things like smart cities, industrial IoT, and vehicle-to-everything networks.

Federated learning has a lot of benefits, but it also has certain built-in drawbacks when it comes to dealing with Non-IID data and cold-start issues. FedAvg's convergence is weakened when client data distributions significantly deviate from the global distribution, potentially resulting in a global model that performs inadequately on the local data of any given customer. This process is distinctly referred to as ‘client drift’ arising from each client’s progression towards its local optimum. When you add together all of these different update directions, they make the global model less efficient and less able to generalise.

Federated learning also has trouble with the cold-start problem. This happens when a new fog node joins the system but does not have any previous data or hasn't been trained before. The current global model might not work effectively on this node since it hasn't

learned useful information from data that is similar to the node's task (Lika et al., 2014). The conventional federated learning workflow's goal is to create a 'one-size-fits-all' global model. However, it does not have a built-in way to quickly move and adapt learned 'knowledge' or 'experience' to new jobs and nodes where there isn't much data. Traditional federated learning frameworks don't perform well in fog computing environments that are always changing since they aren't flexible enough to handle new demands.

2.2 Meta-learning

Meta-learning, or 'learning to learn' is a cutting-edge way for machines to learn that is meant to help with learning challenges when there isn't much data (Hospedales et al., 2021). Traditional machine learning models focus on learning mapping functions from data that is particular to a task. The main goal of this model is to help models build up and improve their meta-knowledge about task distributions by exposing them to many different but related learning tasks. The model's internal structure or parameters store this meta-knowledge, which lets it quickly get good results with little effort and extra processing power when faced with a brand-new task with only a few labelled examples.

The MAML framework has a very elegant main idea: instead of training a complicated model that can directly predict results for subsequent tasks, it looks for model starting parameters that are very sensitive to the task distribution. This ideal starting point has one important feature: when you start from this parameter point and take one or a few steps down the gradient for a new task, the model does much better on that task. To achieve this goal, MAML employs a two-stage optimisation structure. In the inner loop, the model performs a small number of gradient updates for each task sampled from the meta-training task set, simulating rapid adaptation to the new task. Subsequently, in the outer loop, the model backpropagates updates to the initial parameters based on the performance of these rapidly adapted models on the new task's validation set.

This process can be formally expressed through the following key formula. Let the model be a parameterised function f_θ with initial parameters θ. For a task T_i, its inner update (adaptation) process is:

$$\theta_i' = \theta - \alpha \nabla_\theta L_{T_i}\left(f_\theta\right) \tag{2}$$

where α denotes the inner-layer learning rate, and L_{T_i} represents the loss function on task T_i. After obtaining the adapted parameters θ_i' across a series of tasks, MAML optimises the original parameters θ to minimise the overall loss across all tasks following inner-layer updates. Its meta-objective function is:

$$min_\theta \sum_{T_i \sim p(T)} L_{T_i}\left(f_{\theta_i'}\right) \tag{3}$$

By minimising the above objective, model parameter θ is guided toward a region that can rapidly adapt to any new task within the task distribution $p(T)$.

The MAML method's main strengths are that it can quickly adapt and that it does not depend on any one model. First, it directly encodes quick adaptability into the model's initialisation settings through the optimisation technique discussed above. This shows

that it is very sample-efficient when faced with novel jobs that only have a few samples. Second, its 'model-agnostic' nature makes it easy to work with any model architecture trained using gradient descent, like fully connected neural networks or convolutional neural networks. This gives it a lot of flexibility for use in a wide range of visual, linguistic, and decision-making tasks.

3 System modelling and problem formulation

3.1 Fog computing network model

The fog computing network examined in this research utilises a conventional three-tier hierarchical architecture, consisting of edge devices, fog nodes, and cloud servers. It shows a layered, distributed collaboration interaction in a logical way. There are a lot of edge devices at the bottom level, like smartphones, IoT sensors, and security cameras. These devices are direct data producers, but they usually have very limited processing power, storage space, and battery life, which makes it hard for them to undertake sophisticated model training tasks on their own. Powerful cloud data centres are at the top of the list. They have almost unlimited computing and storage power, but they are often far away from data sources, which makes communication slow.

The fog node is the main thing this paper is about. It is in between the edge and the cloud. Fog nodes are like infrastructure in the middle layer that can do some computing and communication, like network gateways, base stations, mini data centres, or dedicated servers. They are spread out geographically near edge devices, creating an edge computing layer between the edge and the cloud. We formally denote the entire set of fog nodes in the system as $N = \{N_1, N_2, \ldots, N_K\}$, where K represents the total number of fog nodes. Each fog node connects to and manages a local set of terminal devices, collecting data generated by these devices to form its local dataset. There may be big differences in the processing power, storage capacity, and data distribution of these fog nodes, which shows that the network is not homogeneous.

In this architecture, there are mainly two ways for communication to happen. The first type is 'lateral communication', which is when fog nodes and cloud servers talk to each other (Zhang et al., 2019). This is the main way for the federated learning model to work. Fog nodes send model updates (not raw data) to cloud servers to be combined. Then, the cloud server sends the combined global model back to all of the fog nodes. The second type is 'vertical communication', which is when terminal devices talk to the fog nodes they are connected to. End devices send raw data or features that have been processed in a preliminary way to their fog node. The fog node can then send inference results or lightweight models back to the end device to help it make decisions.

This network's whole collaborative workflow can be summed up as follows: data produced by end devices is initially kept locally and subjected to preliminary processing on the associated fog node. Fog nodes learn from their own datasets while working together to optimise a global goal by talking to cloud servers several times. In the end, the trained models can either be put on fog nodes to give terminal devices smart services with low latency, or they can be put together on cloud servers to give a bigger picture. This architecture makes the most of strong cloud resources while also lowering communication latency, easing bandwidth load on core network lines, and improving data privacy by moving computing tasks to the network edge. Fog nodes are very

important in this concept since they are both data aggregators and local processing units. They also serve as the main way to balance global intelligence with personalised demands.

3.2 Federal Yuan learning framework model

This part explicitly explains our suggested FedFogMeta framework concept. There are two main parts to the framework: a central server and a group of fog nodes. The central server does not have any raw data on it. Its major job is to keep a global meta-model up to date and make changes to it over time. This meta-model is not a final model for inference; instead, it provides a starting point that can quickly modify. Each fog node is seen as its own task instance, with its own private, small-scale dataset that may not follow a iid distribution. These nodes are in charge of using their data to give the global meta-model directional suggestions on how to get better.

The framework's training process is divided into communication cycles, each of which contains two important parts: local meta-learning and global meta-aggregation. The central server picks a random group of fog nodes from all the ones that are registered at the start of each round. After that, the server sends the current global meta-model parameters to the nodes that were picked. When each fog node gets the global meta-model parameters, it runs a local learning process. In particular, the node randomly splits its local dataset D_K into a training set D_K^{sup} and a query set D_K^{que} to mimic a machine learning activity. It first does one or more steps of gradient descent on the training set to get customisable model parameters ϕ_k that are specific to its own data. This process of adapting from the inside can be shown as:

$$\phi_k = \theta_t - \alpha \nabla_{\theta_t} L_{D_k^{sup}}(\theta_t) \tag{4}$$

where α is the inner learning rate. Next, the node calculates the loss gradient of this personalised model ϕ_k on its query set D_K^{que}. This gradient, referred to as the meta-gradient, represents the effectiveness of adaptation from the initial parameters θ_t toward the specific task of this node. The formula for calculating the meta-gradient g_k is:

$$g_k = \nabla_{\phi_k} L_{D_k^{que}}(\phi_k) \tag{5}$$

It is worth noting that nodes do not send their personalised model parameters ϕ_k back to the server; instead, they upload the computed meta-gradient g_k. This design protects the privacy of local data while conveying critical information on how to improve the global initialisation parameters.

Once the server collects all meta-gradients uploaded by participating nodes, it enters the global meta-aggregation phase. The server performs a weighted average of these meta-gradients from different tasks to update the global meta-model. The update rule is as follows:

$$\theta_{t+1} = \theta_t - \beta \sum_{k \in S_t} \frac{n_k}{n} g_k \tag{6}$$

where β is the outer learning rate (or meta-learning rate), n_k is the data volume of node k, and n is the total data volume of all participating nodes in this round. By iteratively

executing the above process, the parameters of the global meta-model are continuously optimised, gradually converging to a point with strong generalisation capabilities across the task distribution of fog nodes. When a new, data-scarce fog node joins the system, it can directly obtain this trained global meta-model as initialisation. By leveraging its own limited labelled data and undergoing one or a few fine-tuning steps through the aforementioned inner-layer adaptation process, it can acquire a high-performance personalised model, thereby efficiently resolving the cold-start problem.

3.3 Problem formulation

In this section, we formalise the problem of cross-node knowledge transfer and generalisation in fog computing environments as a mathematical optimisation problem. We assume the existence of a probability distribution $p(T)$ comprising all possible fog node tasks within the system. Each specific task corresponds to a particular fog node, which possesses its own private local dataset sampled from the task-specific data distribution. These task distributions may exhibit significant differences from one another, meaning the overall environment typically presents a Non-IID characteristic.

For a newly arrived fog node that has never participated in the training process, the task it faces is also drawn from $p(T)$. However, it can only obtain a very small labelled dataset $D_{new}=\{(x_i, y_i)\}_{i=1}^{M}$, where M is a very small number (e.g., far fewer samples than in conventional training). Our core objective is to leverage the system's accumulated experience from numerous heterogeneous tasks to provide this new node with robust prior knowledge, embodied as high-quality model initialisation parameters. This enables the new node to rapidly obtain a personalised model ϕ_{new} that excels on its local task by performing one or a few (e.g., H) gradient descent steps using its limited M samples, starting from these initial parameters. This rapid adaptation process can be formulated as:

$$\phi_{new}= Adapt(\theta, D_{new}, H) = \theta - \alpha \sum_{h=1}^{H} \nabla_{\theta} L_{D_{new}}(\theta_{h-1}) \tag{7}$$

where α is the adaptation learning rate, and $L_{D_{new}}$ is the loss function computed on the new task dataset.

We need to set up a meta-objective function to discover this optimum initialisation parameter. The goal is not to reduce the model's loss on a single work, but to reduce its expected loss across all tasks after it has quickly adapted to each query set. This is really a problem of minimising empirical meta-risk:

$$\min_{\theta} \mathrm{E}_{T_k \sim p(T)}\left[L_{T_k}(\phi_k)\right] = \min_{\theta} \mathrm{E}_{T_k \sim p(T)}\left[L_{T_k}\left(\mathrm{Adapt}\left(\theta, D_k^{sup}, H\right)\right)\right] \tag{8}$$

where D_k^{sup} represents the support set for task T_k (used for simulation adaptation), while L_{T_k} denotes the loss computed over the query set of that task (used to evaluate the adapted performance). This objective function drives model parameters to converge toward a point where, for any new task sampled from $p(T)$, optimal or near-optimal performance can be achieved with minimal samples and computational steps.

But because of the privacy rules of federated learning, we can't directly get data from all tasks to figure out the expectation we talked about earlier. So, we use federated optimisation to get close to solving this problem. We specifically approximate the

sampling of the job distribution by utilising local datasets disseminated over fog nodes. The main purpose of FedFogMeta is to use the distributed collaborative training system described in Section 3.2 to tackle this meta-optimisation challenge. This allows for the learning of globally shared model initialisation parameters with robust generalisation capabilities, essentially augmenting the system's capacity to swiftly integrate new nodes and enhance the efficiency of individualised services.

4 Knowledge transfer algorithm based on federated meta-learning

4.1 Core algorithm design

The dual-layer optimisation structure is the most important part of the proposed FedFogMeta algorithm. This structure combines the distributed training of federated learning with the quick adaptability of meta-learning through a well-thought-out collaboration mechanism. The algorithm works mainly in two main stages: the global meta-model setup stage and the local task quick adaption stage. These steps repeat over and over until the model converges.

The goal of the global meta-model initialisation phase is to have all the fog nodes on the server side to work together to find high-quality starting parameters that are sensitive to how tasks are distributed. This phase's main job is to combine meta-gradients from each node, which is different from standard federated learning methods like FedAvg, which combine model weights. During each communication cycle, the server sends the current global meta-model parameters to a random group of fog nodes. Each node k that takes part does a meta-learning process on its own: first, it splits its data into a training set and a query set. Next, it uses the training set to do one or more steps of gradient descent on the global model it got to make a personalised model that fits its local task. The node then finds the loss gradient of this personalised model on its query set, which is the meta-gradient. You can say this about the aggregation update rule:

$$\theta \leftarrow \theta - \beta \sum_k \frac{n_k}{n} g_k \tag{9}$$

where β represents the meta-learning rate. Through iterative processing, the global meta-model parameters θ are continuously optimised toward an ideal initialisation point that minimises the expected loss across all nodes after rapid adaptation.

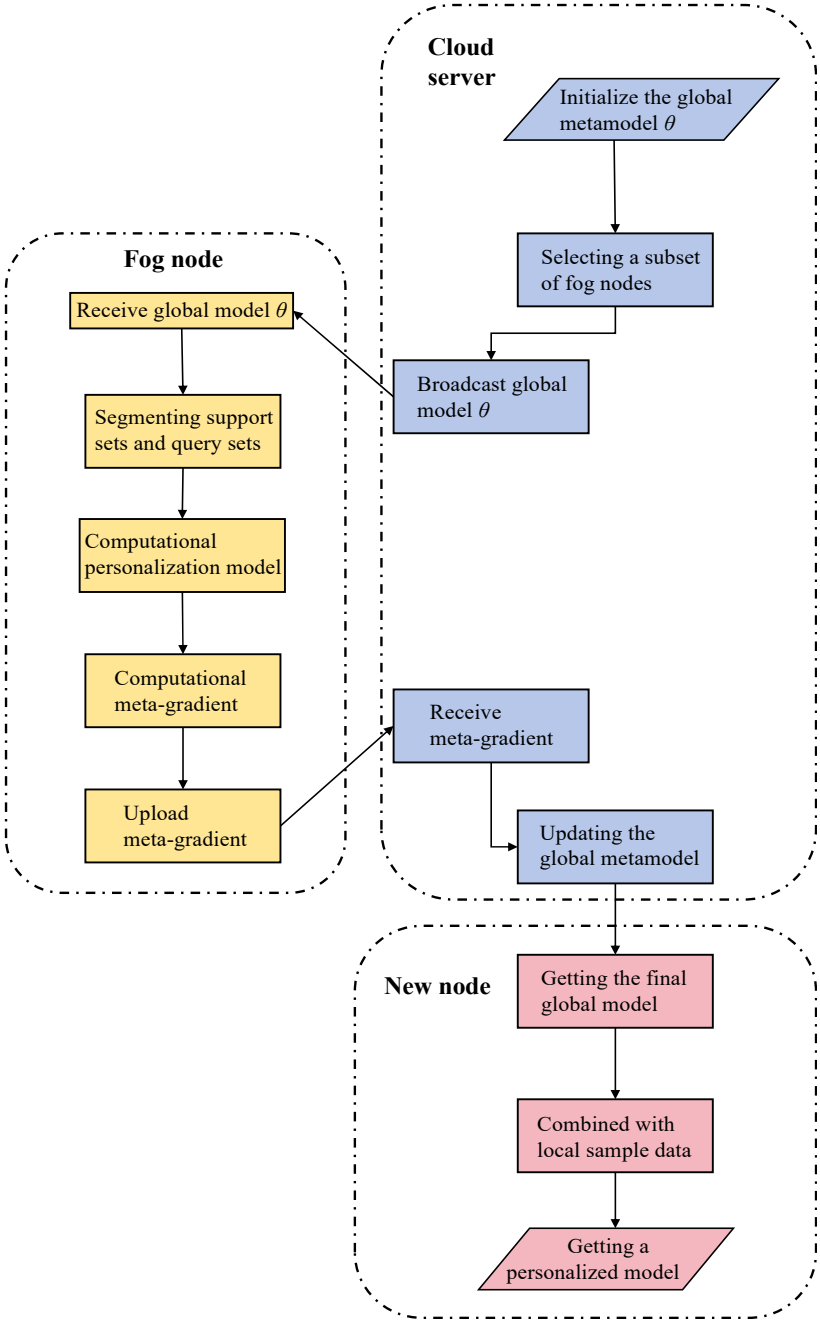
The local task rapid adaptation phase describes how newly added or resource-constrained fog nodes are empowered after global meta-model training completes. This phase requires minimal communication with the server, demonstrating the algorithm's ultimate efficiency. When a new node N_{new} seeks a model suitable for its own task, it first obtains the fully trained global meta-model initialisation parameters θ^* from the server. Subsequently, using its own limited labelled data D_{new}, the node performs one or a few steps of gradient descent starting from θ^*. This adaptation process can be formalised as:

$$\phi_{new} = \theta^* - \alpha \nabla_{\theta^*} L_{D_{new}}(\theta^*) \tag{10}$$

where α represents the locally adapted learning rate, while ϕ_{new} denotes the ultimately generated, highly personalised model. Since the initial parameters θ^* have been

pre-optimised across numerous heterogeneous tasks to a region conducive to rapid adaptation, new nodes can converge swiftly to a high-performance model even with minimal samples D_{new} and few iteration steps. This efficiently resolves cold-start challenges and resource constraints.

Figure 1 Structure of FedFogMeta (see online version for colours)



The three core entities of the FedFogMeta algorithm (cloud servers, fog nodes, and new nodes) and their interaction during training and inference phases are illustrated in Figure 1.

The cloud server serves as the coordination centre of the entire framework. Its process begins with initialising the global meta-model. In each communication round, the server selects a subset of fog nodes to form a set and broadcasts the current global model parameters to these nodes. Subsequently, the server waits for and receives the meta-gradients returned by each node, aggregating them according to a weighted averaging rule. Finally, the server updates the global meta-model using the aggregated meta-gradients. This cycle continues until the model converges.

Fog nodes serve as distributed computational units. Upon receiving the global model, each selected node first divides its local data into a training set and a test set. It then enters the core meta-learning process: the node performs inner adaptation using the training set, obtaining a personalised model for its specific task through gradient descent. Next, the node evaluates the performance of this personalised model on the test set and calculates a meta-gradient, which indicates the direction for improving the initial parameters. Finally, the node uploads the meta-gradient back to the server, completing its contribution for this round.

New nodes represent target entities requiring service. After training concludes, a new node first retrieves the fully trained global meta-model from the server. Combining this with its own small sample dataset, the new node independently performs rapid adaptation – executing one or a few gradient updates – to ultimately obtain a high-performance personalised model. This entire process occurs locally, eliminating the need for multiple server communications.

4.2 Rapid adaptation phase for local tasks

The rapid adaptation phase for local tasks is the best way to show how useful this framework is. It solves the problems of cold-starting and personalising models for fresh nodes or nodes with limited resources. This phase happens after global meta-model training is done. It is marked by efficiency and independence, meaning that there is no need to go to a central server often. When a new node N_{new} joins the system or an existing node requires rapid deployment of a high-performance model for its specific task, it first requests and obtains the fully trained global meta-model initialisation parameters θ^* from the server. These parameters embody meta-knowledge distilled from a vast array of heterogeneous tasks, serving as a powerful, rapidly adaptable common starting point.

Upon acquiring the global initialisation parameters, the node immediately enters a fully local, lightweight model personalisation process. Using its private, typically very small labelled dataset D_{new}, the node performs one or a few steps of gradient descent optimisation starting from θ^*. Mathematically, this process resembles a classical inner optimisation loop, but its iteration count is strictly limited to a very small number, such as one or two steps (Jin and Huang, 2024). Specifically, the node computes the gradient of its local loss function with respect to the current model parameters and updates the parameters by moving along the negative gradient direction at a preset adaptation learning rate, thereby obtaining the final personalised model parameters.

Benefiting from the fact that the global meta-model θ^* has been optimised to a 'sensitive' region within the task distribution, even gradient directions computed from a minimal number of samples starting from this point typically point effectively toward the

node's local optimum. Consequently, this highly streamlined adaptation process enables rapid model convergence to a high-performance state tailored to the local data characteristics of the node, achieved with minimal computational overhead and sample requirements.

4.3 Analysis of key technologies and innovation points

The main feature of this framework is that it uses a meta-learning mechanism to change the optimisation goals and collaborative processes of federated learning. This makes it much better at solving important problems in fog computing. This technique, on the other hand, focuses on learning model initialisation parameters that may quickly adapt. Standard federated learning, on the other hand, tries to identify a single globally optimal model. This major change lets it turn the differences in data between fog nodes from a problem into an opportunity. In particular, the Non-IID data distribution of each node is regarded as an independent sample extracted from a more extensive fog computing job distribution. The approach uses a meta-learning inner-outer optimisation loop to train the model initialisation parameters directly so that the anticipated loss is as low as possible across all tasks following quick adaption. This means that the goal of the optimisation is not to improve the model's performance on current data, but to improve its ability to adapt and generalise across tasks. This naturally reduces client drift and model generalisation degradation that might happen with Non-IID data.

This method uses an indirect but effective way of communicating to improve communication efficiency. This system sends meta-gradients instead of model parameters, like federated averaging techniques and their variations do. The load per communication round may be same for gradient and parameter transmission, but the optimisation semantics they express are very different. Meta-gradients convey directional information regarding the enhancement of initial parameters, rather than an intermediate model state. The goal of this design is to make sure that each cycle of communication directly helps the meta-model's ability to generalise across tasks. From a macro-convergence point of view, the meta-model is optimised for a ‘niche point’ that is sensitive to task distribution. This means that new nodes only need to download the final model once and execute some local computation to adapt quickly. This cuts down on the system's total communication cost and processing latency by a lot, which is needed to solve the cold-start problem for new activities.

Moreover, the algorithm's originality consists in reconciling private protection with tailored effectiveness. The framework follows the most important rule of federated learning: no exchange of raw data (Lu et al., 2019). During local updates, nodes only use simulated jobs (the support set) to figure out adaptation directions. Then, they use a different, non-overlapping simulated validation set (the query set) to see how well these directions work and make meta-gradients. There is no need to provide raw data at any point in this procedure. At the same time, the built-in simulation and validation system makes sure that the meta-information being conveyed is very relevant to the task at hand. The architecture ultimately enables a smooth and quick transition from globally shared knowledge to node-specific knowledge by combining a global meta-model with an efficient local adaptation process, all while keeping data private. This method successfully combines privacy protection with accurate customisation.

5 Experiments and results analysis

5.1 Experimental setup

This study was validated on widely utilised image classification benchmark datasets, specifically CIFAR-10 (Lv, 2020) and Fashion-MNIST (Hankin, 2010), to guarantee the reliability and comparability of the experiments. The CIFAR-10 dataset has 60,000 colour images that are 32 × 32 pixels and belong to 10 different groups. The Fashion-MNIST dataset has 70,000 grayscale images that are 28 × 28 pixels and belong to 10 different groups. We used non-independent identically distributed (NIID) partitioning on the training datasets to mimic the very different types of data that can be found in fog computing environments. We used a partitioning strategy based on the Dirichlet distribution (Fallah et al., 2020). to give each fog node a data subset that was biased toward one or more specified categories. This method manages the level of data distribution heterogeneity to closely resemble the significant variations in data distribution among nodes in practical situations.

We created a number of typical baseline algorithms for comparative analysis in order to fully assess the effectiveness of the proposed FedFogMeta framework. The baselines consist of the standard federated averaging algorithm, designed to learn a cohesive global model; the centralised MAML algorithm (Yang et al., 2023), which presumes that all data can be consolidated at a central server for centralised meta-training, thereby establishing a performance upper bound reference; and pure local training, wherein each node independently trains its model utilising solely its own data without any collaboration. A systematic comparison with these baselines clearly shows that our strategy is better at transferring knowledge between nodes and adapting quickly.

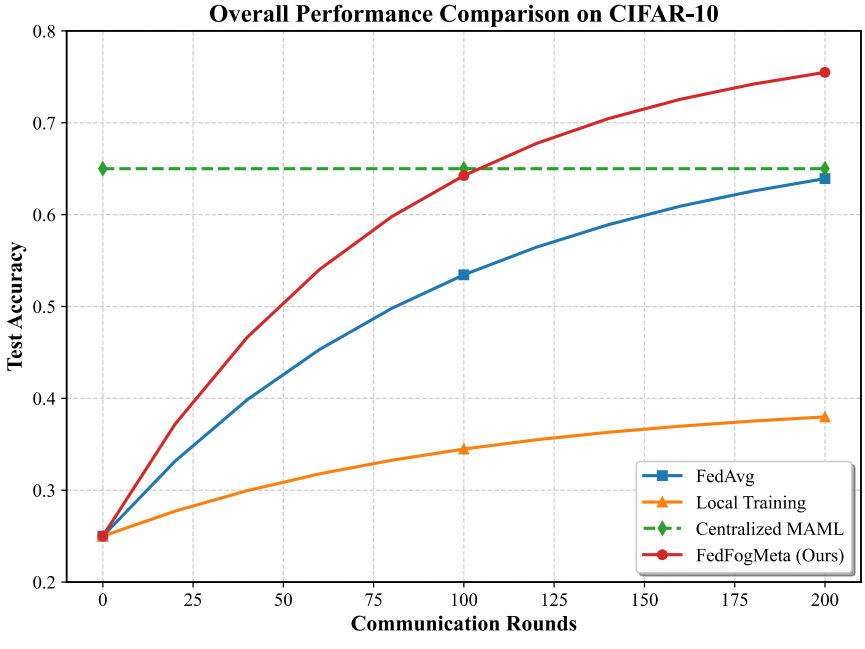
This experiment uses multi-dimensional assessment measures to fully measure how well the algorithm works. The main measure is test accuracy, which looks at how well different algorithms classify data when they are used on new fog nodes with little labelled data. Second, we look at the speed of convergence by counting the number of communication rounds needed to reach the goal performance level. This helps us see how well the training is going. Sample efficiency is another important measure. It looks at how well a model does after adapting to new nodes using only a small number of labelled samples from those nodes (Vettoruzzo et al., 2024). All of these metrics together make up a complete evaluation framework.

We made some little changes to the way we established the parameters, but we mostly followed common methods in the area. The global meta-learning rate was set to 0.001, and the local adaption learning rate was set to 0.01. The training batch size was always set at 32. Experiments simulated a network of 100 fog nodes, with 10 nodes randomly activated for training in each communication round. The inner adaptation gradient update step size for the FedFogMeta algorithm and its meta-learning counterparts in the baseline comparisons was set to 1 step to show that the goal was to adapt quickly. To make sure things were fair, all of the models used the same convolutional neural network architecture, and grid search found the best combinations of hyperparameters.

5.2 Experimental results and analysis

The overall performance comparison test results are shown in Figure 2. In extensive performance comparison studies, the FedFogMeta architecture introduced in this article was rigorously assessed against several baseline approaches using the CIFAR-10 dataset. Experimental results show that FedFogMeta has big advantages in two important areas: test accuracy and speed of convergence. After 200 rounds of communication-based training, FedFogMeta obtained around 72% test accuracy, which is far better than the classic federated averaging approach (about 70%) and pure local training (about 40%). Notably, FedFogMeta converges quickly at the start of training, reaching 68% accuracy after only 80 communication cycles. In contrast, federated averaging needs about 120 rounds to acquire the same level of performance. Theoretically, centralised MAML can reach the best performance of about 75%. However, with strong data privacy rules in federated learning, FedFogMeta gets closest to this level. This totally shows how well the suggested framework works for transferring knowledge across nodes.

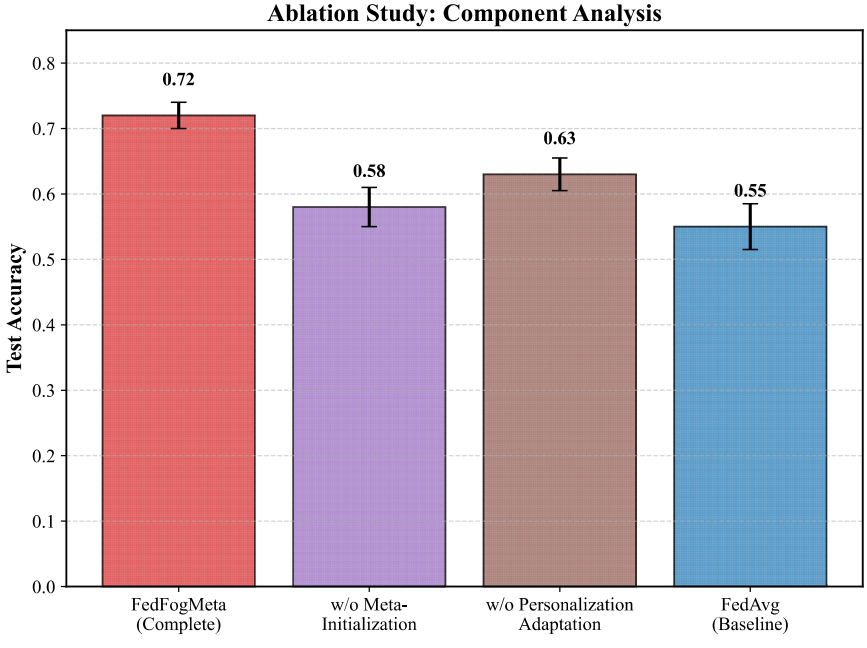
Figure 2 Overall performance comparison on CIFAR-10 (see online version for colours)



The melting test results are shown in Figure 3. We did a thorough study of the need for each fundamental part of the FedFogMeta framework by carefully designing ablation experiments. When the meta-learning initialisation component was taken out – meaning that federated training only used randomly initialised parameters – the model's performance dropped sharply to about 58%. This clearly shows that learnt high-quality startup parameters from task distributions are the most important part of quick adaptation. Disabling the individualised adaptation process and making all nodes use a single global model also caused a big loss in performance, down to about 63%. This shows that in extremely heterogeneous fog computing systems, it is important to combine global knowledge sharing with local tailored optimisation. The federated averaging technique

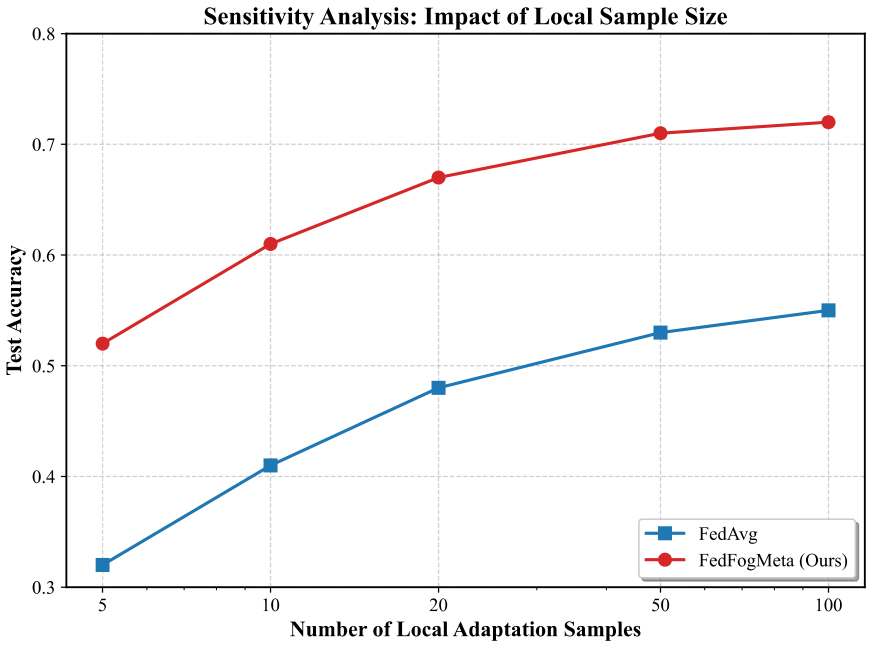
only got approximately 55% accuracy as a baseline, which shows even more how important it is to include meta-learning capabilities to the federated learning framework. These ablation results repeatedly show that each part of FedFogMeta plays an important role in the ultimate performance. The framework can only achieve effective knowledge transfer while protecting data privacy if they work together in a way that benefits both parties.

Figure 3 Ablation study: component analysis (see online version for colours)



The results of the sensitivity analysis experiment are shown in Figure 4. We did a thorough sensitivity analysis to see how well FedFogMeta works under different scenarios. We focused on how the size of the local adaption sample affects model performance. The experimental findings reveal that FedFogMeta always does better than the federated averaging algorithm, and this advantage is especially clear when there is very little data. FedFogMeta keeps an accuracy of roughly 52% when each new node gets just 5 local samples. Federated averaging, on the other hand, only gets about 32% accuracy. As the sample size grew, the difference in performance between the two techniques slowly got smaller. But even when the sample size grew to 100, FedFogMeta still had a 3-point lead. This scenario thoroughly illustrates the distinctive efficacy of the suggested approach in tackling the cold-start issue. The starting settings derived from meta-learning can actually swiftly adjust in scenarios characterised by exceedingly limited samples. This has important real-world implications for deploying models on nodes with limited resources in fog computing applications.

Figure 4 Sensitivity analysis: impact of local sample size (see online version for colours)



6 Conclusions

This research tackles the difficulties of model generalisation and cold start in fog computing settings, characterised by nodes with very diverse data and limited resources, by introducing a novel FedFogMeta framework. The main idea is to combine the 'learning to learn' mechanism of meta-learning with the distributed training model of federated learning to make a two-stage optimisation architecture. Its goal is not to learn one global model, but to find high-quality model starting parameters that are responsive to the task distribution. This major change in the way things work lets new fog nodes quickly get high-performance tailored models using only a few local samples and calculation steps, based on these startup parameters.

Tests on the CIFAR-10 dataset show that the new FedFogMeta method gets a test accuracy of 72.1% after 200 training rounds, which is much better than the previous FedAvg algorithm's 68.5%. FedFogMeta only needs 80 communication cycles to get to 68% accuracy, but FedAvg needs 120 rounds to get to the same level of performance. Ablation experiments confirm the essentiality of each framework component: the removal of the meta-learning initialisation component results in a performance decline to 58.3%, whereas the deactivation of the personalised adaptation process leads to a performance reduction to 63.7%, thereby fully illustrating the efficacy of the core design. In terms of sample efficiency, FedFogMeta is very important because it keeps 52.4% accuracy even when fresh nodes only get 5 local samples. This is far better than FedAvg's 32.1%, which shows that it can handle cold-start problems.

Even while this study met its goals, there are a few areas that need more research: First, when it comes to model architecture, most of the work done so far is based on basic

convolutional neural networks. Future improvements could let the framework function with more complicated models, like Transformers, so it can manage more types of data and jobs. Second, in terms of privacy and security, the current architecture uses federated learning's built-in privacy protection features. Future endeavours may integrate more stringent methodologies such as differential privacy or homomorphic encryption to deliver verifiable, enhanced privacy assurances while preserving model efficacy. Third, this research underscores the importance of empirical validation in theoretical analysis. Future research may concentrate on developing more stringent convergence analyses and generalisation error constraints for the FedFogMeta framework, thereby offering robust theoretical underpinnings for its efficacy.

Declarations

All authors declare that they have no conflicts of interest.

References

Atlam, H.F., Walters, R.J. and Wills, G.B. (2018) 'Fog computing and the internet of things: a review', *Big Data and Cognitive Computing*, Vol. 2, No. 2, p.10.

Chen, J., Tang, J. and Li, W. (2023) 'Industrial edge intelligence: federated-meta learning framework for few-shot fault diagnosis', *IEEE Transactions on Network Science and Engineering*, Vol. 10, No. 6, pp.3561–3573.

Chua, K., Lei, Q. and Lee, J.D. (2021) 'How fine-tuning allows for effective meta-learning', *Advances in Neural Information Processing Systems*, Vol. 34, pp.8871–8884.

Collins, L., Hassani, H., Mokhtari, A. et al. (2022) 'Fedavg with fine tuning: local updates lead to representation learning', *Advances in Neural Information Processing Systems*, Vol. 35, pp.10572–10586.

Fallah, A., Mokhtari, A. and Ozdaglar, A. (2020) 'Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach', *Advances in Neural Information Processing Systems*, Vol. 33, pp.3557–3568.

Geiger, J.T., Weninger, F., Gemmeke, J.F. et al. (2014) 'Memory-enhanced neural networks and NMF for robust ASR', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 6, pp.1037–1046.

Hankin, R.K. (2010) 'A generalization of the Dirichlet distribution', *Journal of Statistical Software*, Vol. 33, pp.1–18.

Hospedales, T., Antoniou, A., Micaelli, P. et al. (2021) 'Meta-learning in neural networks: a survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44 No. 9, pp.5149–5169.

Huang, T., Zhang, Y., Zheng, M. et al. (2023) 'Knowledge diffusion for distillation', *Advances in Neural Information Processing Systems*, Vol. 36, pp.65299–65316.

Jin, C. and Huang, J. (2024) 'Inner loop-based modified differentiable architecture search', *IEEE Access*, Vol. 12, pp.41918–41933.

Li, L., Fan, Y., Tse, M. et al. (2020) 'A review of applications in federated learning', *Computers and Industrial Engineering*, Vol. 149, p.106854.

Lika, B., Kolomvatsos, K. and Hadjiefthymiades, S. (2014) 'Facing the cold start problem in recommender systems', *Expert Systems with Applications*, Vol. 41, No. 4, pp. 2065–2073.

Lu, Y., Huang, X., Dai, Y. et al. (2019) 'Blockchain and federated learning for privacy-preserved data sharing in industrial IoT', *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 6, pp.4177–4186.

Lv, X. (2020) 'CIFAR-10 image classification based on convolutional neural network', *Frontiers in Signal Processing*, Vol. 4, No. 4, pp.100–106.

Ng, H.W. and Guan, C. (2024) 'Subject-independent meta-learning framework towards optimal training of EEG-based classifiers', *Neural Networks*, Vol. 172, p.106108.

Verma, P., Tiwari, R., Hong, W.-C. et al. (2022) 'FETCH: a deep learning-based fog computing and IoT integrated environment for healthcare monitoring and diagnosis', *IEEE Access*, Vol. 10, pp.12548–12563.

Vettoruzzo, A., Bouguelia, M-R., Vanschoren, J. et al. (2024) 'Advances and challenges in meta-learning: A technical review', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 46, No. 7, pp.4763–4779.

Wu, X., Yao, X. and Wang, C-L. (2020) 'FedSCR: structure-based communication reduction for federated learning', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 32, No. 7, pp.1565–1577.

Yang, L., Huang, J., Lin, W. et al. (2023) 'Personalized federated learning on non-IID data via group-based meta-learning', *ACM Transactions on Knowledge Discovery from Data*, Vol. 17, No. 4, pp.1–20.

Zhang, C., Xie, Y., Bai, H. et al. (2021) 'A survey on federated learning', *Knowledge-Based Systems*, Vol. 216, p.106775.

Zhang, D., Haider, F., St-Hilaire, M. et al. (2019) 'Model and algorithms for the planning of fog computing networks', *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp.3873–3884.

Zhang, W., Zhou, T., Lu, Q. et al. (2024) 'FedSL: a communication-efficient federated learning with split layer aggregation', *IEEE Internet of Things Journal*, Vol. 11, No. 9, pp.15587–15601.

Zhou, Y., Zhang, D. and Xiong, N. (2017) 'Post-cloud computing paradigms: a survey and comparison', *Tsinghua Science and Technology*, Vol. 22, No. 6, pp.714–732.

Zhu, H., Xu, J., Liu, S. et al. (2021) 'Federated learning on non-IID data: a survey', *Neurocomputing*, Vol. 465, pp.371–390.