The Journal of Supercomputing
https://doi.org/10.1007/s11227-024-06232-4

Cluster knowledge-driven vertical federated learning

Zilong Yin[1] · Xiaoli Zhao[1] · Haoyu Wang[1] · Xin Zhang[1] · Xin Guo[3] · Zhijun Fang[2]

Accepted: 14 May 2024

Abstract

In industrial scenarios, cross-departmental collaboration is necessary to achieve quality traceability. However, data cannot be shared due to privacy concerns. *Vertical Federated Learning* (VFL) enables heterogeneous industrial sectors to jointly train models while preserving product privacy. However, existing traditional VFL algorithms only focus on aligning feature benefits and suffer from high communication costs and poor performance. This paper proposes a "*Cluster Knowledge-Driven Vertical Federated Learning*" (Cluster-VFL) algorithm, which integrates cluster intelligence to optimize heterogeneous distributed environments. In Cluster-VFL, each participant engages in training as an individual within the cluster, taking into account the utilization of non-aligned features. Cluster-VFL promotes model updates by identifying global optimal individuals and transferring global optimal knowledge. Subsequently, this knowledge is merged with the individual optimal knowledge obtained from local training of each participant. We conducted extensive experiments using an open-source diagnostic dataset and a proprietary dataset from Company A. The results unequivocally demonstrate that this algorithm enhances participants' learning abilities, improves their communication efficiency.

✉ Xiaoli Zhao
evawhy@163.com

Zilong Yin
zilong_yin@163.com

Haoyu Wang
wanghy@sues.edu.cn

Xin Zhang
kxamanda@163.com

Xin Guo
smilegx110@hotmail.cn

Zhijun Fang
zjfang@dhu.edu.cn

[1] Shanghai University of Engineering Science, 333 Longteng Road, Shanghai 201620, China

[2] Donghua University, 2999 North Renmin Road, Shanghai 201620, China

[3] Sanda University, 2727 Jinhai Road, Shanghai 201209, China

Published online: 30 May 2024

Keywords Vertical federated learning · Cluster intelligence · Knowledge-driven

1 Introduction

In the industrial manufacturing process, products undergo various departments such as design, craft, production, and detection [1, 2]. Quality concerns may arise during product usage [3], necessitating collaborative efforts among these departments for quality traceability [4, 5]. However, data sharing becomes impractical due to privacy constraints and departmental interests [6]. *Vertical Federated Learning*(VFL) offers a solution by accommodating heterogeneous data sources, enabling the training of machine learning models without compromising the confidentiality of raw data [7, 8].

In *Federated Learning* (FL), the parameter exchange after each round of local training results in extensive parameter communication, especially when deployed on IoT and edge devices with limited communication capabilities [9–11]. This situation creates a conflict between frequent parameter communication and the limited communication capabilities of the devices, making communication a bottleneck in Federated Learning, consequently affecting learning efficiency and performance. Unlike *Horizontal Federated Learning* (HFL), which allows communication for parameter aggregation after multiple rounds of participant training, VFL necessitates parameter exchange participation in every round of gradient aggregation, exacerbating the challenge of communication costs in VFL. To address this issue, various methods have been proposed to reduce communication overhead and enhance learning efficiency. Quantization methods [12] involve converting high-precision parameters into low-precision values for communication, while sparsification techniques [13, 14] focus on reducing data size by discarding redundant parameters to improve communication efficiency. In contrast to these methods, pruning methods [15, 16] assess the update status of the entire model after each training round and subsequently adjust the communication frequency of each node based on the evaluation results.

However, these methods have brought to light a pivotal challengs. While they primarily concentrate on assessing the training outcomes of individual local devices as the cornerstone for enhancing communication efficiency and performance in federated learning, the true essence of federated learning lies in collaborative training to attain a shared network model. Therefore, a mere focus on the training results of each node falls short in optimizing communication efficiency and performance. Instead, a holistic consideration of the interplay between local and global models is imperative to achieve a more effective optimization of communication efficiency and performance [17].

Swarm intelligence embodies the notion that cluster knowledge can yield optimal solutions in distributed environments [18]. Certain HFL algorithms have also corroborated the adaptability of this concept to heterogeneous data environments within federated settings [19, 20]. These algorithms selectively involve only a subset of participants during aggregation rounds, reducing the aggregation rounds and accelerating the extraction of learning from local models. This demonstrates that similar concepts can concurrently enhance both the learning capabilities of federated models and communication efficiency. In HFL, participants need only align sample entities, with swarm intelligence primarily addressing the performance deficiencies in models caused by limited alignment of sample entities. In VFL, both features and samples need alignment, exacerbating the challenge of data heterogeneity. Therefore, in theory, swarm intelligence principles are also applicable to VFL.

In this paper, we propose a cluster knowledge-driven VFL algorithm (Cluster-VFL). In Cluster-VFL, each unlabeled client within the cluster participates as an individual in the VFL process. We adopt a method of full-feature alignment, leveraging labeled clients as communication mediators to extend features with zero padding, ensuring that each participant possesses features missing in their own dataset but present in others,' thus ensuring consistency in the model architecture. Labeled clients construct local network layers with fixed network weights, seamlessly integrating with unlabeled client models. The labeled client model computes expected values for the network docking layer, distributing them to unlabeled clients as pseudo-labels. Individual optimal knowledge is computed by unlabeled clients, which is then collected by labeled clients and assessed for global optimality before dissemination. Finally, unlabeled clients update their local models based on individual and global optimal knowledge. The specific contributions of this algorithm are as follows:

1. We propose a method of full-feature alignment, integrating all features from participating clients into model training. This approach not only considers aligning shared features but also extends to encompass all features from all participating clients, aiming to address potential feature omission and data imbalance inherent in traditional methods. Furthermore, it aims to enhance the model's learning capacity and accelerate convergence speed.
2. We incorporate cluster knowledge into VFL and propose a novel approach that integrates individual and global knowledge to update the local models of VFL participants. This method aims to enhance communication efficiency and the learning capability of subpar models. By combining individual and global knowledge, this integration enables the search for the global optimum solution and facilitates the sharing of high-quality knowledge.
3. Cluster-VFL achieves optimal results on two industrial datasets: the open-source dataset Diagnosis and dataset A from a certain automotive company.

2 Related work

This section will cover the previous studies that have influenced the proposed algorithm in this paper, including "Federated Learning," "Horizontal Federated Learning in Cluster Knowledge" and "Vertical Federated Learning."

2.1 Federated learning

Federated Learning (FL) is a distributed machine learning paradigm that enables clients (devices or organizations) to collaborate in training machine learning models without exposing the clients' local data [21]. In recent years, significant research progress has been made in addressing the challenges of *Horizontal federal learning*

(HFL). FedProx [22] addresses the limitations of traditional FedAvg [21] in handling device and data heterogeneity by introducing a proximal term constraint to mitigate model bias caused by data heterogeneity. FedAsync [23] tackles the issue of regularization locality by adjusting the weights of outdated models using a stiff function when updating the global model to ensure convergence. FedMSA [24] proposes a federated learning model selection and adaptation system, which includes a hardware-aware model selection algorithm, supporting full automation in building and deploying fl tasks to different hardware. Virtual Homogeneity Learning [25] and Data-Free Knowledge Distillation [26] leverage virtual homogeneous datasets to alleviate data heterogeneity in FL. Bias-Variance [27] reduces the bias and variance of local gradients by considering the heterogeneity of local objects. KD3A [28] constructs transferable consensus knowledge by distilling knowledge from models trained on multiple source domains. However, while these methods address data heterogeneity and performance, they overlook the collaborative nature of FL and communication challenges. Therefore, focusing solely on the training results of local clients is insufficient. Instead, it is essential to consider the interaction between local and global models comprehensively to optimize communication efficiency and performance effectively.

2.2 Horizontal federated learning in cluster knowledge

Previous research has proposed solutions to the communication challenges in FL. One approach is to increase the number of local training rounds and reduce the communication rounds by aggregating intermediate parameters in unstable network environments [21]. However, due to the diversity of network structures, the amount of intermediate parameters transmitted in each communication round also increases accordingly. FedSGD [29] reduces the number of communication parameters by aggregating intermediate gradients. However, this significantly increases the number of communication rounds. Neither of these algorithms effectively addresses the communication problem. To address this issue, researchers have integrated cluster knowledge, namely local and global collaboration patterns, into FedAvg. For example, FedPSO [19] uses particle swarm optimization to update the global model, requiring clients to only send loss values to the server. In return, the server requests model parameters from only one client each time, reducing the amount of communication between clients and servers and improving model performance. Similarly, FedSA [20] employs a simulated annealing strategy to select hyperparameters and participant subsets for aggregation in each round. By optimizing hyperparameters related to the convergence of the global model, this approach reduces the aggregation rounds and accelerates convergence. In terms of image classification, [30] proposes a cluster-based FL strategy that maximally reduces total model error and includes control mechanisms to minimize cooperative attacks. This strategy introduces cluster knowledge and adjusts aggregation methods based on model quality. It has been proven that these three strategies incorporating cluster knowledge effectively utilize the interaction between clients and servers to reduce communication, accelerate convergence, and improve FL performance. Their effectiveness has also been validated in various industrial scenarios, such as water monitoring systems and stroke prediction, where AquaFeL-PSO [31] and FL-PSO [32] have been successfully applied. However, existing methods are all based on improvements in HFL, while VFL scenarios with larger feature differences and more unlabeled clients impose significant limitations on such cluster knowledge.

2.3 Vertical federated learning

The integration of cluster knowledge-driven learning with VFL faces greater challenges in practical industrial scenarios compared to the mature combination of cluster knowledge thinking with HFL. Classical VFL algorithms such as AggVFL [33] and SplitVFL [33], lacking labels, require communication for every round of gradient updates. Additionally, labeled clients need to synthesize gradients using compound partial derivatives, significantly increasing computational and communication costs. To address these issues, FedBCD [34] allows each party to perform multiple local updates before each communication round, effectively reducing communication overhead. FedMVT [35] and FedCVT [36] predict pseudo-labels for unlabeled samples by estimating representations of missing features, and finally jointly train three classifiers using different sources or views of the extended training set to filter out invalid samples. While existing VFL works are effective, the collaborative learning mode of VFL under cluster knowledge-driven paradigms may still make breakthroughs in communication and performance.

3 Problem definition

We address a scenario in VFL involving two types of clients. One type, referred to as unlabeled clients, possesses only sample features, while the other, labeled clients, holds both sample features and labels. There exists some overlap in the data between the two clients in terms of samples and features. Figure 1 illustrates the data distribution for both clients in this scenario.

Specifically, x^u and x^l denote the sample features of the unlabeled and labeled clients, respectively, with n^u and n^l features each. y^l represents the sample labels of the labeled client. There exists a partial overlap between the two, comprising m^{al} samples and n^{al} features. Among these m^{al} overlapping samples, there are $n^{u,nl}$ and $n^{l,nl}$ non-aligned feature data in each.

Traditional VFL utilizes the overlapping data of size $m^{al} \times n^{al}$ to construct a federated machine learning model, which results in a considerable waste of non-aligned feature data. Additionally, challenges such as communication costs associated with the update patterns during each round of gradient aggregation and performance enhancements have been inherent issues in VFL. Therefore, our proposed method of full-feature alignment aims to address the feature omission problem in traditional VFL. By integrating the cluster intelligence into VFL and distributing global knowledge, the aim is to reduce the frequency of communication. Simultaneously, the integration of individual and global knowledge in training is aimed at enhancing the performance of the federated model.

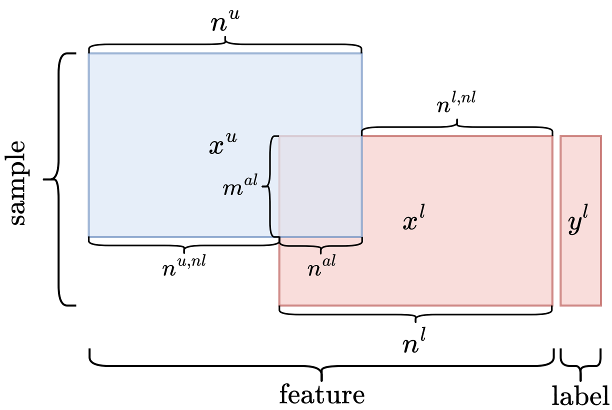


Fig. 1 Virtual views of datasets in VFL

4 Proposed algorithm

In this paper, we propose a cluster knowledge-driven VFL algorithm, termed Cluster-VFL. which consists of three phases:

- **Full-Feature Alignment** aims to address the issue of feature omission in traditional VFL. By integrating all features from participating clients into model training, even those absent in certain clients are filled with zero values. This approach not only aligns shared features but extends to encompass all features from all participating clients, ensuring each party's full contribution to model training.
- **The Expected Values of the Shared Network Docking Layer** refer to the anticipated outputs generated by the labeled client model for the network docking layer. In VFL, the labeled client constructs local network layers with fixed network weights. These layers are seamlessly integrated with the models of unlabeled clients. The labeled client model computes the expected values for the network docking layer, which are then distributed to unlabeled clients as pseudo-labels. These expected values serve as predictions or target outputs for the unlabeled clients' training data, guiding their model updates during the VFL process.
- **Cluster Knowledge-Driven Innovation** is a novel approach in VFL where individual optimal knowledge is computed by unlabeled clients. This knowledge is then gathered by labeled clients and evaluated for global optimality before being disseminated. Ultimately, unlabeled clients update their local models based on both individual and global optimal knowledge. This innovative method leverages the collective intelligence of the cluster to enhance model performance and convergence in VFL scenarios. By combining individual and global perspectives, Cluster Knowledge-Driven Innovation fosters a more comprehensive and effective approach to model training.

The architecture of the proposed algorithm is shown in Fig. 2, which includes the algorithm introduction 4.1 and the communication process 4.2. The algorithm introduction consists of full-feature alignment 4.1.1, the expected values of the shared network docking layer 4.1.2 and cluster knowledge-driven innovation 4.1.3. In this section, we will provide a detailed description of the aforementioned parts.

4.1 Algorithm introduction

4.1.1 Full-feature alignment

We propose a novel approach to feature alignment in industrial manufacturing. In this context, various departments handle different components, each possessing distinct features. These differences in features significantly impact the learning ability of VFL. Therefore, our approach aims to mitigate the impact of these differences on VFL by employing feature completion techniques.

In Cluster-VFL, full-feature alignment is a necessary condition driven by cluster knowledge. In this method, unlabeled clients first encrypt their local features and transmit the feature count and their positions to the labeled client. Subsequently, the labeled client sorts the feature names, assigning a custom identifier as a key to each feature and retaining the feature names as values. This mapping is encapsulated within a dictionary

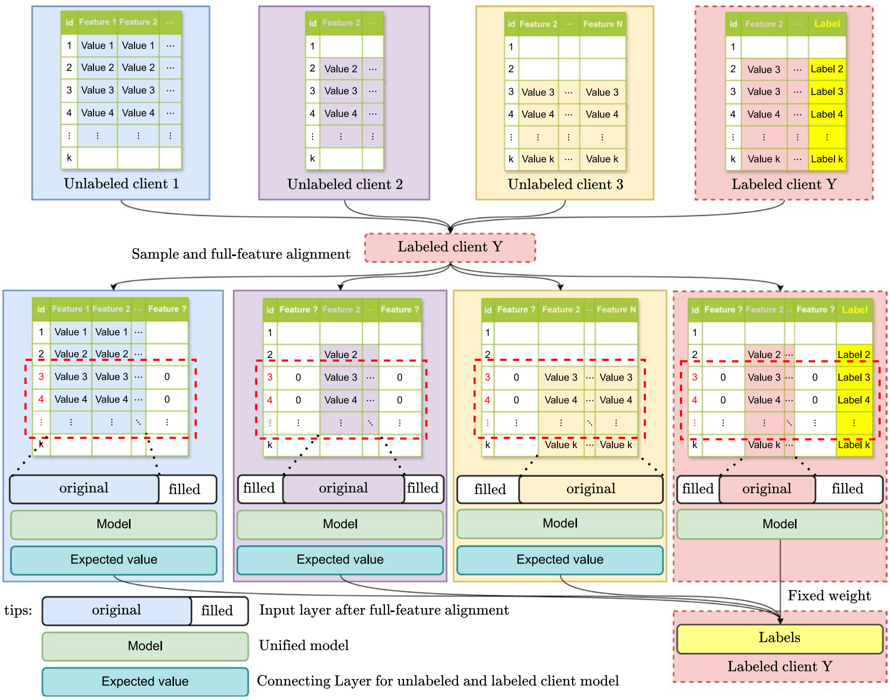


Fig. 2 The architecture of Cluster-VFL

and sent back to the unlabeled clients. The unlabeled clients then adjust the order of their local data features according to the dictionary, filling any gaps with zero values to ensure the participation of all clients in VFL. Furthermore, as cluster knowledge-driven training involves each client in the cluster training based on individual and global optimal knowledge, each client has the potential to become the global optimum. Therefore, unlabeled clients must possess a unified model architecture to meet the prerequisite for knowledge sharing.

4.1.2 The expected values of the shared network docking layer

In the Cluster-VFL framework, due to the absence of labels for unlabeled clients, the network weights of the labeled client remains constant throughout the entire model. Various adjustments can be made to protect the security of labeled data and provide pseudo-labels to unlabeled clients. These adjustments may involve altering the mapping rules for label encoding, expanding the dimensions of encoding, or scaling the encoding values, followed by the application of homomorphic encryption. During the communication interval between labeled and unlabeled clients, unlabeled clients conduct local training based on pseudo-labels and record a set of model parameters in each training iteration. Subsequently, each unlabeled client selects the model with the highest accuracy as its local individual optimal knowledge. Upon completing local training, each unlabeled client sends the corresponding accuracy value associated with this optimal knowledge to the labeled client. The labeled client compares and selects the client with the highest accuracy, then request its model parameters. Finally, these parameters are distributed to all unlabeled clients as the global optimal knowledge.

4.1.3 Cluster knowledge-driven innovation

During each round of parameter sharing in Cluster-VFL communication, unlabeled clients update their model cluster knowledge using both their local optimal individual knowledge and the global optimal knowledge obtained from labeled clients through multiple rounds of local training. This integrated approach significantly reduces communication costs from unlabeled to labeled clients while achieving the goal of sharing cluster knowledge. The updating process can be summarized as follows:

$$V_{ij}(t+1) = \omega_t V_{ij}(t) + c_p \gamma_p \left(X_{(\text{ pbest})(j)}(t) - X_{ij}(t) \right) + c_g \gamma_g \left(X_{(\text{ gbest})(j)}(t) - X_{ij}(t) \right) \quad (1)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (2)$$

In formula 1 and 2, ω represents the cluster inertia weight. $X_{ij}(t)$ and $V_{ij}(t)$ represent the value and change of the jth model parameter at the i-th unlabeled party during the t-th communication. $X_{(\text{ pbest })(j)}(t)$ and $X_{(\text{ gbest })(j)}(t)$ represent the jth parameter of the individual's best knowledge and the j-th parameter of the globally shared best knowledge, respectively, during the t-th communication. γ_p and γ_g are random values between 0 and 1. c_p and c_g represent the individual and global learning factors.

The cluster knowledge-driven approach leverages both individual optimal knowledge and global optimal knowledge to aggregate and update. However, due to the heterogeneous distribution of unlabeled data, the global optimal knowledge may not be suitable for all unlabeled instances. When training using VFL, dynamically setting rules for dual knowledge aggregation is beneficial for better utilization of cluster knowledge-driven approach. We have improved the update strategy for cluster knowledge-driven approach by initially setting a large individual learning factor γ_g and a small global learning factor γ_p, which weakens the cluster experience while improving the self-awareness of unlabeled instances, thereby increasing cluster diversity. In the later stages, we set a small individual learning factor γ_p and a large global learning factor γ_g to improve global optimization capability. The updated adjustment rules for the cluster knowledge-driven approach are as follows:

$$\omega(t) = \omega^{\max} - \frac{t * \left(\omega^{\max} - \omega^{\min}\right)}{t^{\max}} \tag{3}$$

The influence of parameter changes from the previous round of unlabeled clients on the current round of parameter changes, or the degree of trust expressed by the unlabeled clients in the current level of parameter changes, is represented by the parameter ω, which is an inertia weight and a highly significant parameter in cluster knowledge. In formula 3, ω^{max} and ω^{min} represent the maximum and minimum values of ω, while t and t^{max} stand for the current and maximum number of communications, respectively. ω decreases linearly. Initially, a larger value of ω facilitates global diffusion search within each feasible solution space for unlabeled clients, thereby preventing premature convergence to local optima caused by the self-search of unlabeled clients. Later on, a smaller value of ω improves the search capability within the optimal solution range suited for the unlabeled clients.

$$c_p(t) = c_p^{\min} + \frac{t * \left(c_p^{\max} - c_p^{\min}\right)}{t^{\max}} \tag{4}$$

$$c_g(t) = c_g^{\max} - \frac{t * \left(c_g^{\max} - c_g^{\min}\right)}{t^{\max}} \tag{5}$$

In formula 4 and 5, the rates of updating for individual optimal and global optimal parameters are represented by coefficients c_p and c_g, respectively. These coefficients demonstrate linearly increasing and decreasing trends. Unlike traditional particle swarm optimization strategies, VFL initially utilizes the guidance of global optimal parameters to help inferior clients search for global optimal solutions, thereby maximizing the effectiveness of these parameters. In the later stages of iteration, clients adaptively search within the refined search range of the global optimal solutions to find the most suitable solution for themselves, thereby maximizing the effectiveness of individual optimal parameters. The cluster knowledge-driven approach does not seek a global optimal solution through training. Instead, it enables the use of global optimal knowledge to guide each unlabeled client in finding adaptive optimal solutions suitable for themselves.

4.2 Communications process

Unlike traditional VFL, which utilizes the intermediate results of gradients for communication, Cluster-VFL employs the cluster's global optimal knowledge for communication. This reduces the frequency of communication. The communication process depicted in Fig. 3 will be described in this section.

(1) In Cluster-VFL, the first round of communication involves the unlabeled clients sending their homomorphically encrypted feature names ID_f and sample numbers ID_s to the labeled party. The labeled party then assigns indexes based on the order of obtaining non-repeating feature names and stores the mapping relationship between indexes and feature names in dictionary *dict*. Afterward, all samples are aligned to obtain ID_s', as shown in the pseudocode 1.

Algorithm 1 Labeled client receive

```
function LABELED CLIENT RECEIVE(ID_f, ID_s)                        ▷ (1)
    for feature in ID_f do
        if feature not in dict then
            dict[len(dict)] ← feature.
        end if
    end for
    ID'_s ← ID_s.intersection(ID'_s).
end function
```

(2)* The labeled party sets up a contingency matrix according to the number of label categories, which serves as the fixed weights for the network. (2) It aligns local samples and labels based on ID_s'. The expected value of the docking layer between the labeled and unlabeled network is calculated as y'. Finally, the labeled party sends the unified network architecture *NN* and feature name dictionary *dict* to the participants, as shown in pseudo code 2.

Algorithm 2 Labeled client send

```
function LABELED CLIENT SEND                                       ▷ (2)
    ID_s ← ID'_s.
    y ← y[ID'_s].
    staggered_matrix ← random(eye(category)).                     ▷ (2)*
    y' ← y * staggered_matrix.
    return y', dict, NN, ID'_s.
end function
```

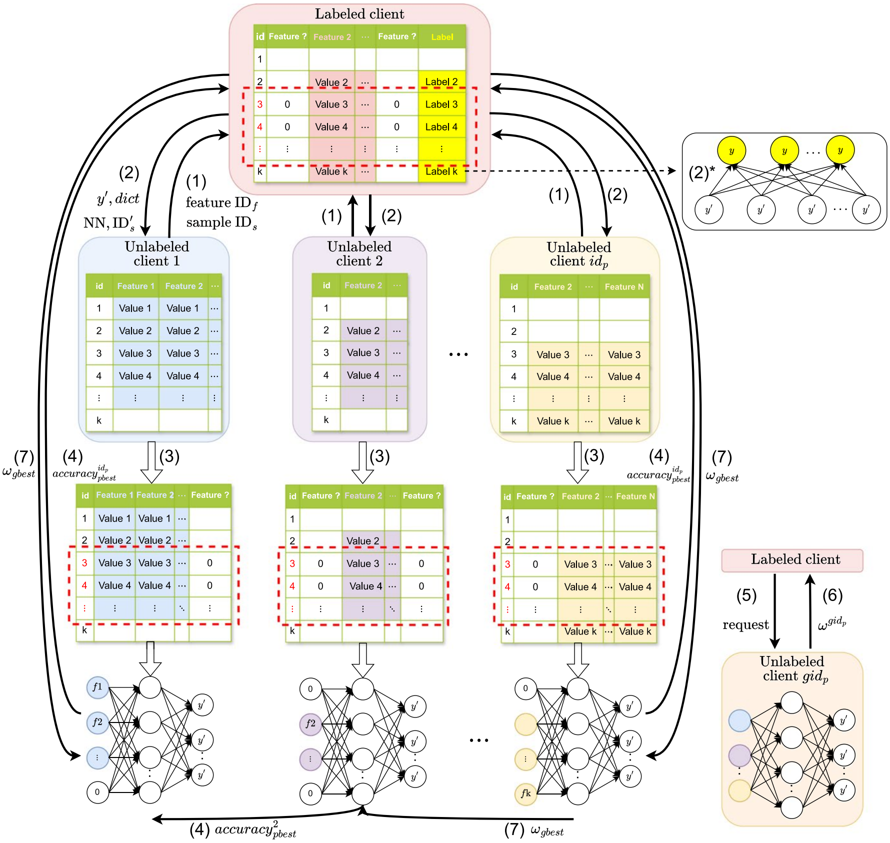


Fig. 3 Communication process for Cluster-VFL. ID_s represents Sample' ID. ID_f represents Feature' encrypted name. ID'_s represents Align Sample' ID. NN represents Shared Model Structure. y' represents labels for network encryption. *dict* represents dictionary of mapping relationships between feature'ID and encrypted feature names. $accuracy^{id_p}_{pbest}$ represents individual optimal accuracy with unlabeled client id_p. gid_p represents unlabeled client's index with individual optimal accuracy. ω^{gid_p} represents individual optimal parameters with unlabeled client id_p. ω_{gbest} represents ω^{gid_p}

(3) Unlabeled clients align with local samples using ID'_s and utilize y' as pseudo-labels. Our own dictionary supplements missing features with a value of 0. Every unlabeled party constructs the same network using NN (see pseudocode 3).

Algorithm 3 Unlabeled client receive

function UNLABELED CLIENT RECEIVE($y', dict, NN, \mathrm{ID}'_s$) ▷ (3)
  $\mathrm{ID}_s \leftarrow \mathrm{ID}'_s$.
  $y' \leftarrow secret\_y[id_p]$.
  for feature in $dict$ **do**
    if feature not in ID_f **then**.
      $\mathrm{ID}_f.insert(\mathrm{ID}_?)$ and fill $\mathrm{ID}_?$ with 0 values.
    end if
  end for
  Build a model based on NN.
end function

During formal training in Cluster-VFL, each unlabeled party serves as a member of the cluster and learns through the sharing of globally optimal knowledge from a labeled party. After each unlabeled party completes several rounds of local training, (4) the accuracy rate $accuracy^{id_p}_{pbest}$ of individual optimal knowledge is sent to the labeled party. The labeled party compares these evaluation values and selects the optimal value, (5) which is then sent to the holder $gid\ p$ of the optimal value through a request. (6) The unlabeled clients $gid\ p$ send their individual optimal knowledge ω^{gidp}_t to the labeled party. (7) The labeled party distributes this parameter as the globally optimal knowledge to the unlabeled clients. Finally, the unlabeled clients update their local cluster knowledge-related parameters (ω, c_p, c_g) based on formulas 3, 4, and 5, and then update their local models based on formulas 1 and 2, driven by cluster knowledge. This process repeats through several local iterations until the termination condition is met, as shown by the pseudocode 4.

Algorithm 4 Cluster-VFL

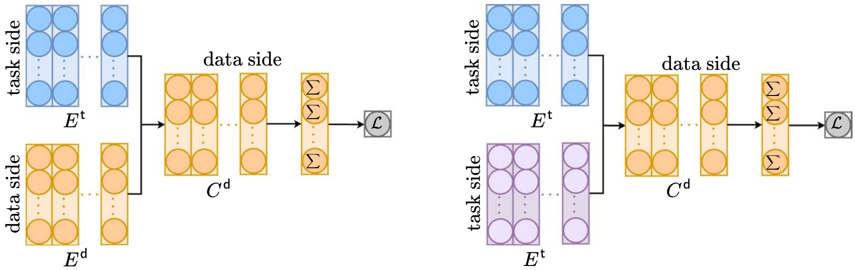
for $id_p \leftarrow 1$ to pp_num **do**
  LABELED CLIENT RECEIVE(fID_{id_p}, sID_{id_p}). ▷ (1)
end for
$y', dict, NN, \mathrm{ID}'_s \leftarrow$LABELED CLIENT SEND. ▷ (2)
for $id_p \leftarrow 1$ to pp_num **do**
  UNLABELED CLIENT RECEIVE($y', dict, NN, \mathrm{ID}'_s$). ▷ (3)
end for
while $t < t^{max}$ **do**
  for $id_p \leftarrow 1$ to pp_num **do**
    if $t \neq 1$ **then**
      Correction ω, c_p, c_g based on formula 3,4,5.
      Update NN parameters based on formula 1,2.
    end if
    Train the local NN to send $accuracy^{id_p}_{pbest}$ to labeled client. ▷ (4)
  end for
  Labeled client compares $accuracy^{id_p}_{pbest}$, and the owner ID of the max value is gid_p. ▷ (5)
  Labeled client sends a request to unlabeled client gid_p. ▷ (6)
  Unlabeled client sends $\omega^{gid_p}_t$ to labeled client. ▷ (7)
  Labeled client sends ω_{gbest} to all unlabeled client clients.
end while

5 Experiment

5.1 Experimental setup

5.1.1 Baseline

Based on the training process of VFL, it can be subdivided into two training modes: Split Vertical Federated Learning (SplitVFL) and Aggregated Vertical Federated Learning (AggVFL). In the images in this section, the unlabeled client is named t-square, and the labeled client is named d-square.



(a) Split Vertical Federated Learning(SplitVFL) (b) Center Split Vertical Federated Learning (SplitVFL$_c$)

Split Vertical Federated Learning(SplitVFL): Split learning is an approach in VFL. As illustrated in Fig. 4a, in SplitVFL, each unlabeled client executes a deep learning model adapted to its own dataset. This model serves as the first half of the overall model, ensuring consistency in output format at the final layer, thus enabling aggregation of results from different unlabeled clients. Aggregation can take various forms such as simple averaging or weighted averaging, combining outputs from different unlabeled clients to generate intermediate results of the overall model. These intermediate results are then passed to the labeled client, who executes the second half of the overall model to generate the final output.

In the training process of SplitVFL, due to the nature of split learning, the model consists of both the task-side model and the data-side model. The parameters of the task-side model typically require collaborative efforts between the unlabeled client and the labeled client. Local gradients are computed separately by each side, and the parameters are trained using composite derivatives. The gradient calculation for the overall model parameters is as follows:

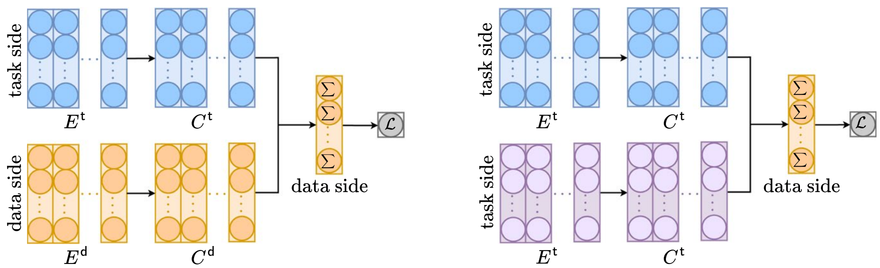
$$\frac{\partial \mathcal{L}}{\partial \omega_a} = \frac{\partial \mathcal{L}}{\partial o_a} \frac{\partial o_a}{\partial \omega_a} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial \omega_p} = \frac{\partial \mathcal{L}}{\partial o_a} \frac{\partial o_a}{\partial o_p} \frac{\partial o_p}{\partial \omega_p} \tag{7}$$

where $\mathcal{L}$ represents the loss function, ω_a and o_a are the model parameters and model outputs of the labeled client, respectively, while ω_p and o_p represent the model parameters and model outputs of the unlabeled client.

As shown in Fig. 4b, a variant of SplitVFL known as SplitVFL$_c$ exists, where the labeled client's dataset lacks feature information and only contains label information, acting as a centralized entity in such scenarios.

SplitVFL offers the advantage of allowing unlabeled clients to learn from the model parameters and results aggregated from the labeled client's posterior model and the final aggregated model output. This theoretically enhances model performance, and different unlabeled clients can design structurally different but output-consistent front-end models according to their needs. However, SplitVFL has the disadvantage of requiring the joint participation of unlabeled clients and labeled client in the training process and model deployment. This increases the computational burden on the labeled client and introduces inconvenience.



(c) Aggregated Vertical Federated Learning (AggVFL)

(d) Center Aggregated Vertical Federated Learning(AggVFL$_c$)

Aggregated Vertical Federated Learning(AggVFL): Aggregation learning, another approach in VFL, operates as depicted in Fig. 4c. In AggVFL, each unlabeled client executes a deep learning model tailored to its dataset, generating uniformly formatted model outputs. These outputs are then transmitted to the labeled client for aggregation, typically through simple averaging or weighted averaging. The aggregated output is used by the labeled client for loss calculation, completing the algorithm's forward computation.

In the backward training process of AggVFL, since the model only resides on the unlabeled client while the model outputs are aggregated by the labeled client, collaboration between task and labeled clients is still required for computing local gradients and training model parameters using composite derivatives. The calculation of model parameter gradients is as follows:

$$\frac{\partial \mathcal{L}}{\partial \omega_p} = \frac{\partial \mathcal{L}}{\partial o_a}\frac{\partial o_a}{\partial o_p}\frac{\partial o_p}{\partial \omega_p} \tag{8}$$

Here, $\mathcal{L}$ represents the loss function, o_a denotes the aggregated model output from the labeled client, and ω_p and o_p are, respectively, the model parameters and outputs from the unlabeled client.

As depicted in Fig. 4d, AggVFL also has a variant known as AggVFL$_c$. In this scenario, the labeled client's dataset lacks feature information and contains only label information, serving as a form of centralized presence.

The advantage of AggVFL lies in the fact that unlabeled clients can design models with different structures according to their needs, as long as the output formats remain consistent. Moreover, aggregation only occurs on the labeled client during the training process, reducing the computational burden on the labeled client. Additionally, once the model is deployed, unlabeled clients can operate independently, enhancing convenience. However, AggVFL's drawback is that during training, learning about the results and experiences of different unlabeled clients is limited to the aggregation performed on the labeled client, which theoretically restricts the model's performance.

5.1.2 Experimental configuration

Device and Simulation Configuration: During the experiment, the following hardware configuration was adopted: CPU 13th Gen Intel(R) Core(TM) i7-13700KF, GPU NVIDIA GeForce RTX 4080, and 32GB RAM. The optimizer Adam was chosen for the training process, with a batch size of 32 and an initial learning rate of 0.001. The values of ω^{max} and ω^{min} were set to 0.3 and 0.1, respectively, while c_p^{max} and c_p^{min} were set to 0.5 and 0. Similarly, c_g^{max} and c_g^{min} were set to 0.5 and 0. The unlabeled model used in this experiment was a 3-layer DNN network. For different dataset sizes and optimization complexities, dataset A had a cluster knowledge-driven iteration count t^{max} of 3 and a local maximum iteration count $epoch^{max}$ of 9. On the other hand, the Diagnosis dataset had a cluster knowledge-driven iteration count t^{max} of 5 and a local maximum iteration count $epoch^{max}$ of 15. As the model parameters gradually approach the global optimum after each cluster knowledge-driven iteration, the optimization difficulty decreases. This leads to the adaptive adjustment of the local iteration count based on formula 9 after updating the unlabeled model with cluster knowledge-driven updates. Consequently, this reduces the response time for the labeled model to wait for the unlabeled model training.

$$epoch = epoch^{max} - \frac{t * epoch^{max}}{t^{max}} \tag{9}$$

Comparative Models: Our algorithm proposes improvements in the communication mode by selecting AggVFL and SplitVFL as comparative models. AggVFL and SplitVFL use traditional gradients for communication. Additionally, we introduce a ablation model called noCluster-VFL. In the context of the previous discussion on "full feature alignment" and "expected values alignment" as sufficient conditions for "cluster knowledge-driven" 4.1, we henceforth denote the method without "cluster knowledge-driven" as "noCluster-VFL." In noCluster-VFL, all unlabeled clients seek the global optimum solely based on their individual optimal knowledge.

5.2 Dataset

This paper uses two datasets: the open-source Diagnosis dataset and the dataset A from a real-life quality traceability scenario in an automotive company.

- *The Sensorless Drive Diagnosis(Diagnosis)* dataset, released by the Institute of Electrical Drives and Machines at the Karlsruhe Institute of Technology in Germany in 2004, consists of data sampled from 21 sensors and includes information on 11 different motor states. The dataset does not include specific information about the features, so machine learning techniques are required to identify the motor's states. The dataset comprises a total of 48,000 samples, which were divided into training and testing sets in an 8:2 ratio.
- *The proprietary dataset from Company A(A)* dataset consists of 40 features and 1 binary label derived from the parts work order across the design, craft, production, and detection departments of a specific car company. The dataset A is comprised of 13,124 samples, and these samples are divided into training and testing sets according to an 8:2 ratio (as shown in Table 1).

Referring to the double deficiency of feature quantity and sample size in VFL data in real-world scenarios (as mentioned in Sect. 1), we simulated a partition of the open-source Diagnosis dataset. We allocate data to unlabeled clients as α times the complete feature quantity and β times the complete sample quantity These control the degree of missing features and samples. A smaller α leads to greater initial feature differences, while a smaller β results in greater initial sample differences. We will evaluate the effectiveness of various algorithms on the pre-segmented Diagnosis dataset and the real-world dataset A.

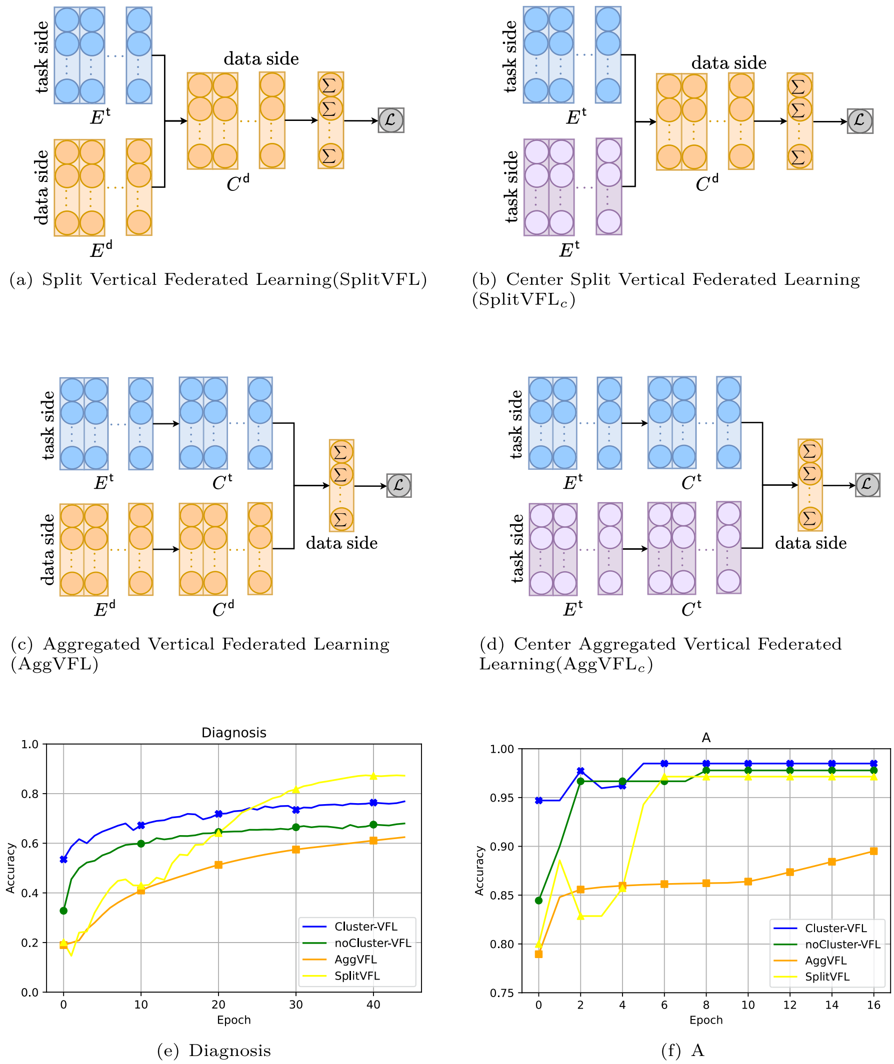
We conducted experiments to compare accuracy and communication efficiency. To simulate a heterogeneous data scenario with missing features and samples, we set parameters α and β to 0.5. The data were divided into three unlabeled clients and one labeled party for the experiments. When evaluating Cluster-VFL, we took into account that the unlabeled clients obtain virtual accuracy data through the expected values in the docking layer. Since the core of cluster-driven knowledge is the optimal participation guided by the participants' knowledge, we calculated the average accuracy data of each party to comprehensively evaluate Cluster-VFL and other comparative models.

5.3 Comparison of accuracy

As shown in Fig. 4, Cluster-VFL achieves early convergence on the diagnosis dataset compared to other algorithms. With the same number of local iterations, Cluster-VFL improves accuracy by 8.89% and 14.41% over noCluster-VFL and AggVFL, respectively. However, due to the large scale of the diagnosis dataset, SplitVFL outperforms in terms of learning ability as it allows labeled clients to independently train local models, thus better optimizing the local models controlled by unlabeled clients. This is advantageous for learning on large datasets, albeit at the cost of increased communication frequency. Therefore, the performance of Cluster-VFL is slightly inferior to SplitVFL. Similarly, on Dataset A, Cluster-VFL also demonstrates earlier convergence compared to noCluster-VFL, AggVFL, and SplitVFL, with accuracy improvements of 0.71%, 8.98%, and 1.34%, respectively, indicating a partial attainment of the optimal solution for industrial participant clusters.

Table 1 Introduction to the dataset A

Department	Feature	Department	Feature
Process/Production/Quality	Production batch	Process/Production/Quality	Item No.
Process/Production/Quality	Part No.	Process/Production	Transmit power
Process/Production/Quality	Component No.	Process/Production	RF frequency
Production	Part sequence No.	Process/Production	Air pressure accuracy
Process/Production	Operation No.	Process/Production/Quality	Temperature accuracy
Quality	Machine code	Process/Production/Quality	Product No.
Process	Test time No.	Process/Production/Quality	Frame count
Process	Chip firmware version	Process/Production	Sensitivity random No.
Process	Chip hardware version	Process/Production/Quality	Version No.
Process/Production	Reference temperature	Process/Production/Quality	Contract No.
Process/Production	Reference pressure	Quality	Inspection worker
Process/Production/Quality	ID range	Quality	Date of inspection
Production	Frame length	Process/Production/Quality	transaction No.
Production	Baud rate	Quality	Inspection start time
Production	Center frequency	Quality	Inspection end time
Production	Battery voltage	Production/Quality	Delivery post
Production	Static acceleration	Production/Quality	retesting
Process/Production/Quality	Hardware platform	Quality	Processing method
Process/Production/Quality	Software version	Quality	Description of processing
Process/Production/Quality	APP version	Process/Production/Quality	Bootloader version
Quality	Determination of quality results		



(a) Split Vertical Federated Learning(SplitVFL)
(b) Center Split Vertical Federated Learning (SplitVFL$_c$)
(c) Aggregated Vertical Federated Learning (AggVFL)
(d) Center Aggregated Vertical Federated Learning(AggVFL$_c$)
(e) Diagnosis
(f) A

Fig. 4 Comparison of accuracy on diagnosis and a dataset using different methods

5.4 Comparison of communication efficiency

In Sect. 4.1.2, it is mentioned that SplitVFL and AggVFL lack labels, requiring intermediate results to be passed with labeled party's data for each gradient update, resulting in establishing communication for each gradient update. However, Cluster-VFL only establishes communication during the initial sharing of feature information and cluster knowledge. Figure 5 illustrates the communication efficiency. Cluster-VFL consistently outperforms other algorithms with the same number of communications on the Diagnosis dataset. On dataset A, Cluster-VFL improves the learning ability of inferior participants by sharing cluster knowledge in the early stages of communication, thereby improving the average accuracy of multiple clients. The experiments demonstrate that Cluster-VFL achieves superior performance in fewer communication rounds, effectively reducing communication costs.

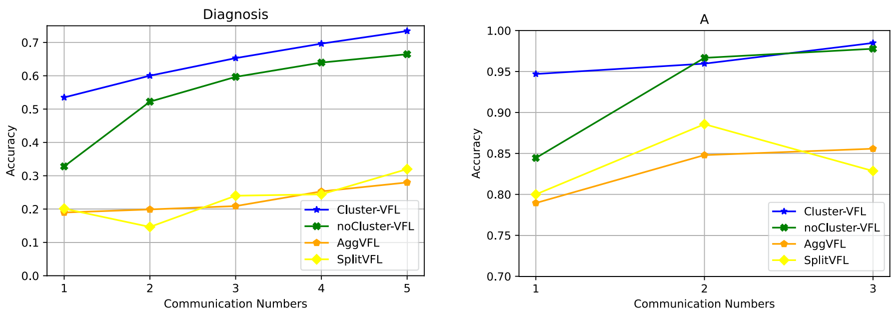
5.5 Ablation experiment

In this section, we will conduct ablation experiments to evaluate the effectiveness of the innovations proposed by Cluster-VFL. The experiments assess two components: the difference in learning ability between using full-feature alignment as model input and using original features as model input, as well as the model optimization capability driven by cluster knowledge.

5.5.1 Comparing of full-feature input and original-feature input in the learning capabilities

In the Cluster-VFL framework, full-feature alignment is a prerequisite for ensuring consistency in the model architecture of participating clients 4.1. To verify the generalizability of this component, we conducted comparative experiments in this section using AggVFL and SplitVFL.

Figure 6 illustrates the accuracy and convergence speed of SplitVFL and AggVFL models on the Diagnosis and Dataset A datasets using the same 30 epochs and consistent random seed, with either full-feature input or original-feature input. Models with full-feature input demonstrate faster convergence speed, optimizing the accuracy-time performance, despite an additional round of local feature-to-server sharing communication before traditional VFL model training. However, this faster convergence facilitates a reduction in communication frequency during subsequent traditional model training, thereby reducing time and communication costs for federated participants in the industrial domain. It is worth noting that models with full-feature input have the same architecture as models with original feature input, except for the input layer.



(a) Diagnosis (b) A

Fig. 5 Comparison of Communication Efficiency on Diagnosis and A Dataset utilizing Different Methods

5.5.2 Comparison of cluster knowledge-driven and non-cluster knowledge-driven in optimal search capability

By contrasting Cluster-VFL with noCluster-VFL, we evaluate the accuracy, and global optimization enhancement of the integration of cluster knowledge-driven and VFL. In noCluster-VFL, clients do not receive global optimal knowledge and only train based on individual optimal knowledge. Before conducting experiments, we set the number of unlabeled clusters to “unlabeled number” and allocate data to unlabeled clients as α times the complete feature quantity and β times the complete sample quantity (mentioned in 5.2). In this study, considering the limited number of features and samples in the complete dataset, we choose the “unlabeled number” values to be 3 and 4. Additionally, to simulate different degrees of heterogeneity in features and samples, we set α and β to 0.9 and 0.5, respectively. A value of 0.9 represents moderate heterogeneity, while 0.5 represents high heterogeneity. Hence, our experiments include four scenarios: high heterogeneity in features, high heterogeneity in samples, high heterogeneity in both features and samples, and moderate heterogeneity in both features and samples.

Table 2 demonstrates that, across varying numbers of participants and degrees of missing features and samples, Cluster-VFL consistently outperforms noCluster-VFL in

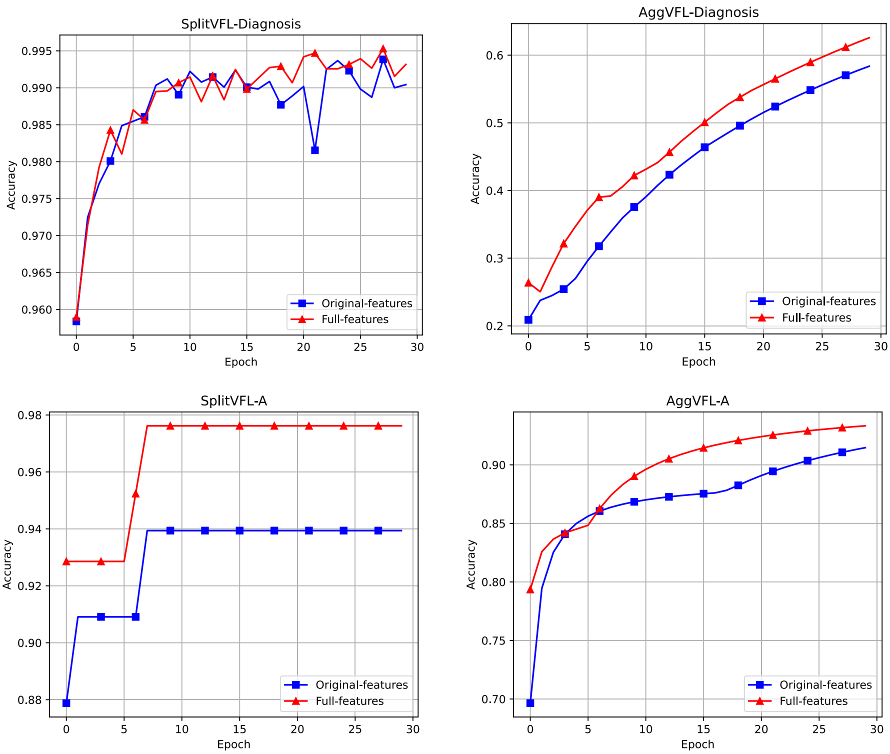


Fig. 6 The Accuracy Comparison of Full-Feature Input After Full-Feature Alignment and Original-Feature Input on Diagnosis and Dataset A utilizing SplitVFL and AggVFL

Table 2 Comparison of accuracy between cluster knowledge-driven and non-cluster knowledge-driven models

Dataset	Unlabeled number	α	β	Cluster-VFL				noCluster-VFL			
				Client1	Client2	Client3	Client4	Client1	Client2	Client3	Client4
Diagnosis	3	0.9	0.9	**0.9930**	**0.9944**	**0.9940**	–	**0.9930**	0.9943	0.9934	–
		0.9	0.5	**0.9904**	**0.9914**	**0.9902**	–	0.9738	0.9703	0.9772	–
		0.5	0.9	0.9845	**0.8352**	**0.5692**	–	**0.9887**	0.8269	0.5670	–
		0.5	0.5	**0.9752**	**0.7696**	**0.5467**	–	0.9538	0.6184	0.4758	–
	4	0.9	0.9	**0.9937**	**0.9950**	0.9938	0.9936	0.9921	0.9944	**0.9944**	**0.9938**
		0.9	0.5	**0.9904**	**0.9897**	**0.9911**	**0.9912**	0.9670	0.9570	0.9699	0.9656
		0.5	0.9	0.9837	**0.8305**	**0.8072**	0.5696	**0.9848**	0.8267	0.8052	**0.5723**
		0.5	0.5	**0.9759**	**0.7873**	**0.7991**	**0.5558**	0.9322	0.5086	0.4873	0.4608

Bold values indicate the highest accuracy for each row terms of client model accuracy. Cluster-VFL has superior capabilities in finding optimal solutions, exhibiting strong optimization performance in performance enhancement, effectively overcoming optimization barriers caused by missing data features and samples. Ultimately, this enables VFL models to more efficiently adapt to industrial edge data.

6 Conclusion

In this paper, we propose a cluster knowledge-driven VFL algorithm (Cluster-VFL), which addresses the challenges of VFL in scenarios where labeled and unlabeled clients coexist. We propose innovative approaches to feature alignment and knowledge driven within this framework. Through full feature alignment driven by cluster knowledge, we ensure the participation of all clients in VFL while minimizing the impact of feature differences. Furthermore, Cluster-VFL enables the efficient distribution of global optimal knowledge to unlabeled clients, enhancing their model performance. By leveraging both individual and global knowledge, Cluster-VFL achieves improved convergence, communication efficiency and performance in VFL settings. Overall, our contributions offer a promising avenue for advancing VFL techniques in industrial manufacturing and other relevant domains. Future research should prioritize improving the global optimization performance of knowledge-driven approaches in VFL datasets from a representational perspective, rather than solely focusing on model parameters. Additionally, exploring the application of federated knowledge-driven approaches to modal, generative problems [37], and beyond would be beneficial.

Acknowledgements The authors would like to thank the anonymous referees for their useful comments. This work is supported by the Science and Technology Innovation 2030 - “New Generation Artificial Intelligence” Major Project, China (2020AAA0109300) and Shanghai Science and Technology Commission, the Shanghai Science and Technology Commission for their generous funding support through project number (22DZ2205600) and Meta-knowledge Driven Multimodal Federated Self-Supervised Learning Algorithm in Intelligent Manufacturing Scenarios (62376151).

References

1. Jinying Li MS, Maiti Ananda, Gray T (2020) Blockchain for supply chain quality management: challenges and opportunities in context of open manufacturing and industrial internet of things. Int J Comput Integr Manuf 33(12):1321–1355. https://doi.org/10.1080/0951192X.2020.1815853
2. Lavelli V, Beccalli MP (2022) Cheese whey recycling in the perspective of the circular economy: modeling processes and the supply chain to design the involvement of the small and medium enterprises. Trends Food Sci Technol 126:86–98. https://doi.org/10.1016/j.tifs.2022.06.013
3. Kwon S, Monnier LV, Barbau R, Bernstein WZ (2020) Enriching standards-based digital thread by fusing as-designed and as-inspected data using knowledge graphs. Adv Eng Inform 46:101102. https://doi.org/10.1016/j.aei.2020.101102
4. Amalfitano D, De Simone V, Maietta RR, Scala S, Fasolino AR (2019) Using tool integration for improving traceability management testing processes: An automotive industrial experience. J Softw Evolut Process 31(6):2171

5. Song C, Wu Z, Gray J, Meng Z (2024) An rfid-powered multisensing fusion industrial iot system for food quality assessment and sensing. IEEE Trans Ind Inform 20(1):337–348. https://doi.org/10.1109/TII.2023.3262197
6. Mulyasari D, Wahyuningtyas R, Alamsyah A (2023) Blockchain technology for privacy protection in healthcare industry. In: 2023 IEEE International Biomedical Instrumentation and Technology Conference (IBITeC), pp. 86–91. https://doi.org/10.1109/IBITeC59006.2023.10390975
7. Liu Y, Kang Y, Zou T, Pu Y, He Y, Ye X, Ouyang Y, Zhang Y-Q, Yang Q (2024) Vertical federated learning: concepts, advances, and challenges. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2024.3352628
8. Huang L, Li Z, Sun J, Zhao H (2024) Coresets for vertical federated learning: regularized linear regression and k-means clustering. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22. Curran Associates Inc., Red Hook, NY, USA
9. Chen S, Yang J, Wang G, Wang Z, Yin H, Feng Y (2024) Clfldp: communication-efficient layer clipping federated learning with local differential privacy. J Syst Arch 148:103067. https://doi.org/10.1016/j.sysarc.2024.103067
10. Ribero M, Vikalo H (2024) Reducing communication in federated learning via efficient client sampling. Patt Recogn 148:110122. https://doi.org/10.1016/j.patcog.2023.110122
11. Liu P, Zhu G, Jiang W, Luo W, Xu J, Cui S (2022) Vertical federated edge learning with distributed integrated sensing and communication. IEEE Commun Lett 26(9):2091–2095. https://doi.org/10.1109/LCOMM.2022.3181612
12. Reisizadeh A, Mokhtari A, Hassani H, Jadbabaie A, Pedarsani R (2020) Fedpaq: a communication-efficient federated learning method with periodic averaging and quantization. Proc Mach Learn Res 108:2021–2031
13. Lin Y, Han S, Mao H, Wang Y, Dally WJ (2018) Deep Gradient Compression: Reducing the communication bandwidth for distributed training. In: The International Conference on Learning Representations
14. Tao Z, Li Q (2018) eSGD: communication efficient distributed deep learning on the edge. In: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). USENIX Association, Boston, MA. https://www.usenix.org/conference/hotedge18/presentation/tao
15. Li W, Wu Z, Chen T, Li L, Ling Q (2022) Communication-censored distributed stochastic gradient descent. IEEE Trans Neural Netw Learn Syst 33(11):6831–6843. https://doi.org/10.1109/TNNLS.2021.3083655
16. Fletcher PT, Venkatasubramanian S, Joshi S (2008) Robust statistics on riemannian manifolds via the geometric median. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. DOI:https://doi.org/10.1109/CVPR.2008.4587747
17. Yang Z, Sun Q (2022) Communication-efficient federated learning with cooperative filter selection. In: 2022 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2172–2176. DOI:https://doi.org/10.1109/ISCAS48785.2022.9937667
18. Dorigo M, Theraulaz G, Trianni V (2021) Swarm robotics: past, present, and future [point of view]. Proc IEEE 109(7):1152–1165. https://doi.org/10.1109/JPROC.2021.3072740
19. Park S, Suh Y, Lee J (2021) Fedpso: federated learning using particle swarm optimization to reduce communication costs. Sensors 21(2):600. https://doi.org/10.3390/s21020600
20. Neto HNC, Dusparic I, Mattos DMF, Fernandes NC (2022) Fedsa: Accelerating intrusion detection in collaborative environments with federated simulated annealing. 2022 IEEE 8th International Conference on Network Softwarization (NetSoft), 420–428
21. McMahan B, Moore E, Ramage D, Hampson S, Arcas BAy (2017) Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Singh, A, Zhu, J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR. https://proceedings.mlr.press/v54/mcmahan17a.html
22. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020) Federated optimization in heterogeneous networks. Proc Mach Learn Syst 2:429–450
23. Xie C, Koyejo O, Gupta I (2019) Asynchronous federated optimization. ArXiv:1903.03934
24. Sun R, Li Y, Shah T, Sham RWH, Szydlo T, Qian B, Thakker D, Ranjan R (2022) Fedmsa: a model selection and adaptation system for federated learning. Sensors 22(19):7244. https://doi.org/10.3390/s22197244
25. Tang Z, Zhang Y, Shi S, He X, Han B, Chu X (2022) Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In: Chaudhuri K, Jegelka S, Song L, Szepesvari C, Niu G, Sabato S (eds.) Proceedings of the 39th International Conference on Machine Learning.

Proceedings of Machine Learning Research, vol. 162, pp. 21111–21132. PMLR. https://proceedings.mlr.press/v162/tang22d.html
26. Zhu Z, Hong J, Zhou J (2021) Data-free knowledge distillation for heterogeneous federated learning. Proc Mach Learn Res 139:12878–12889
27. Murata T, Suzuki T (2021) Bias-variance reduced local sgd for less heterogeneous federated learning. In: Meila, M, Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 7872–7881. PMLR. https://proceedings.mlr.press/v139/murata21a.html
28. Feng H, You Z, Chen M, Zhang T, Zhu M, Wu F, Wu C, Chen W (2021) Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation. In: Meila, M, Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 3274–3283. PMLR. https://proceedings.mlr.press/v139/feng21f.html
29. McMahan H.B, Moore E, Ramage D, Hampson S, Arcas BA (2016) Communication-efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics
30. Połap D, Woźniak M (2021) Meta-heuristic as manager in federated learning approaches for image processing purposes. Appl Soft Comput 113:107872. https://doi.org/10.1016/j.asoc.2021.107872
31. Qolomany B, Ahmad K, Al-Fuqaha A, Qadir J (2020) Particle swarm optimized federated learning for industrial iot and smart city services. In: Global Communications Conference, pp. 1–6
32. Victor N, Bhattacharya S, Maddikunta PKR, Alotaibi FM, Gadekallu TR, Jhaveri RH (2023) Fl-pso: A federated learning approach with particle swarm optimization for brain stroke prediction. 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW), pp 33–38
33. Liu Y, Kang Y, Zou T, Pu Y, He Y, Ye X, Ouyang Y, Zhang Y-Q, Yang Q (2022) Vertical federated learning
34. Liu Y, Zhang X, Kang Y, Li L, Chen T, Hong M, Yang Q (2022) Fedbcd: a communication-efficient collaborative learning framework for distributed features. IEEE Trans Signal Process 70:4277–4290. https://doi.org/10.1109/TSP.2022.3198176
35. Kang Y, Liu Y, Chen T (2020) Fedmvt: semi-supervised vertical federated learning with multiview training. ArXiv:abs/2008.10838
36. Kang Y, Liu Y, Liang X (2022) Fedcvt: semi-supervised vertical federated learning with cross-view training. ACM Trans Intell Syst Technol 13:1–6. https://doi.org/10.1145/3510031
37. Sun R, Zhang Y, Shah T, Sun J, Zhang S, Li W, Duan H, Wei B, Ranjan R (2024) From sora what we can see: a survey of text-to-video generation. arXiv preprint arXiv: