

Article

FedScrap: Layer-Wise Personalized Federated Learning for Scrap Detection

Weidong Zhang ^{1,2} , Dongshang Deng ² and Lidong Wang ^{3,*}

¹ School of Metallurgical Engineering, Anhui University of Technology, Ma'anshan 243002, China; wdzhang@ahut.edu.cn

² School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243002, China; dsdeng@ahut.edu.cn

³ Anhui Institute of Electronic Products Supervision and Inspection, Anhui Information Security Testing Evaluation Center, Hefei 230051, China

* Correspondence: wangld@ahjxw.gov.cn

Abstract: Scrap steel inspection is a critical entry point for connecting the smelting process to the industrial internet, with its security and privacy being of vital importance. Current advancements in scrap steel inspection involve collecting scattered data through the industrial internet, then utilizing them to train machine learning models for distributed classification. However, this detection method exposes original scrap steel data directly to the industrial internet, making it susceptible to interception by attackers, who can potentially obtain sensitive information. This paper presents a layer-wise personalized federated framework for scrap steel detection, termed FedScrap, which leverages federated learning (FL) to coordinate decentralized and heterogeneous scrap steel data while ensuring data privacy protection. The key challenge that FedScrap confronts is the heterogeneity of scrap steel data distributed across the network, which complicates the task of effectively integrating these data into a single detection model constructed via FL. To address this challenge, FedScrap employs a self-attention mechanism to aggregate personalized models for each layer of every client, focusing on the most relevant models to their specific data. By assigning higher attention scores to more relevant models, it achieves more accurate aggregation weights during the model aggregation process. To validate the efficacy of the proposed method, a dataset of scrap images was collected from a steel mill, and the results demonstrate that FedScrap achieves accurate classification of distributed scrap data with an impressive accuracy rate of 90%.

Keywords: scrap steel detection; federated learning; data heterogeneity; network security



Citation: Zhang, W.; Deng, D.; Wang, L. FedScrap: Layer-Wise Personalized Federated Learning for Scrap Detection. *Electronics* **2024**, *13*, 527. <https://doi.org/10.3390/electronics13030527>

Academic Editors: Tao Zhang, Xiangyun Tang, Jiacheng Wang and Jiqiang Liu

Received: 10 January 2024
Revised: 18 January 2024
Accepted: 24 January 2024
Published: 28 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The classification of scrap steel signifies the preliminary phase of the steelmaking process and serves as a crucial component in the integration of heterogeneous industrial internets [1,2]. Currently, the development of scrap steel classification involves utilizing the industrial internet to gather scattered data on scrap steel, subsequently inputting this data into artificial neural networks for classification of the scrap. Although the centralized training via the low-power network [3] facilitates collaboration among scrap data located in various places, it may also give rise to network security and data privacy concerns [4]. The reason is that raw data transmitted directly through the industrial internet may be intercepted by attackers, who could obtain sensitive information to launch network attacks [5,6]. Therefore, a distributed scrap inspection method that can protect the original data is essential.

The emergence of federated learning (FL) makes distributed steel scrap inspection possible. FL trains a machine learning model locally among dispersed scrap data owners, whose parameters are collected to aggregate a shared scrap detection model, which is then fed back to the data owners [7]. Therefore, FL is extremely sensitive to perturbations

of parameters, and a key factor causing perturbations is the distribution of data. The industrial internet is emblematic of a multi-user multiple-input single-output (MU-MISO) heterogeneous network [8]. Its scrap steel data are drawn from numerous recycling points, each with its own distinctive data distribution patterns. For example, industrial scrap recycling prefers a single scrap elimination, while social scrap recycling has a wide variety. The difference of distribution is widely known as the non-independent and identically distributed (Non-IID) problem in FL [9–11]. This kind of Non-IID data makes the model trained by each client biased towards its own data distribution, and the single shared model aggregated by the server cannot represent a unified global distribution, which cannot be generalized to the clients.

In response to the above problems, Personalized Federated Learning (PFL) was proposed to learn a personalized model for the client that has better performance on local data while still benefiting from collaborative training. One way to implement PFL is to cluster clients with similar data distributions, and the model parameters of similar clients are aggregated as their personalized models [12–19]. For example, Yan et al. [12] proposed ICFL, which can dynamically determine the cluster structure of clients during each training round and aggregate a personalized model for each cluster. In [13], Wang et al. proposed CPFL, which uses Earth mover's distance to measure the distribution similarity between client data, thereby clustering clients to combat the Non-IID problem. Long et al. [15] proposed multicenter FL, which forms multiple personalized models through clustering based on distributed similarity of data and optimizes them individually.

The above methods essentially make the client model differentiated and aggregated in the direction of autocorrelation through clustering, and generates the personalized model of client processing Non-IID. However, these cluster-based methods are lacking in the following aspects: (i) The number of partitioned clusters needs to be manually determined, which usually requires first determining the similarity between the data distributions of the clients. (ii) The collaboration between clients of different clusters is interrupted, resulting in a change in the base of model aggregation, but there are few ways to adjust the aggregation weights to accommodate this change. (iii) The object of clustering is the entire model, without considering the similarity relationship between clients at the level of the fine-grained model.

In this paper, we propose a FL framework based on self-attention for scrap classification, which aims to provide a privacy protection and model training platform for distributed scrap recycling with heterogeneous data. The core of self-attention is the self-clustering of clients at the level of granularity in the model, which aggregates a most relevant personalized model for each client, and the weight of the aggregation is adaptively adjusted according to the similarity of the model between clients. In this way, the decentralized scrap recycling sites can combine other inconsistently distributed data to collaboratively train a scrap classification model to meet their personalized needs, without worrying about data privacy leakage. Finally, we collected a scrap classification dataset from steel mills, including 6 scrap varieties, for a total of 1000 samples. The dataset is divided into 10 client datasets with Non-IID characteristics according to the Dirichlet distribution. Specifically, the main contributions of our approach are as follows:

- We have developed a federated steel scrap classification framework based on self-clustering, which allows each client to autonomously aggregate a personalized scrap steel classification model most relevant to them through the self-attention mechanism.
- We propose a model aggregation method based on the self-attention mechanism, which calculates the attention weights between models after serializing the client models, and aggregates personalized models according to the weights.
- We compare the proposed method with multiple personalized FL methods using multiple deep learning models on the dataset collected by ourselves. The experimental results show that the proposed method can effectively improve the classification accuracy of scrap steel under Non-IID distribution.

The structure of this article is as follows: Section 1 provides background and a description of the problem, Section 2 introduces existing methods, Section 3 models the federated scrap steel detection problem, Section 4 presents our solution to the problem, Section 5 validates the effectiveness of the method through experiments, and Section 6 concludes the paper.

2. Related Work

2.1. Scrap Steel Classification

At present, the mainstream classification of scrap steel is to establish a classification model through Deep Learning (DL). In [20], Tu et al. proposed an automatic classification and grading of scrap based on hierarchical learning, which firstly removes complex background information from scrap picture data through the attention mechanism, and then uses a segmentation network to segment the scrap picture. Gao et al. [2] proposed a 3D vision-based scrap steel grading approach, which can detect the edge of thickness features in pictures through machine vision, and use the detected features to classify and grade scrap. Smirnov et al. [21] compared the accuracy of various CNN models for the classification of scrap in railway carriages. Xu et al. [1] used a high-resolution sensor-based image acquisition of scrap steel and proposed a deep learning model based on attention mechanism CSBFNet to classify scrap steel. Williams et al. [22] combined magnetic induction spectroscopy and machine learning technology to develop a scrap classification framework, using magnetic induction spectroscopy to establish the physical characteristics of scrap data as the input of the depth model to classify scrap. DazRomero et al. [23] used principal component analysis to filter scrap out of the environment, and then used deep network DenseNet to classify it.

The classification of non-ferrous metals is also on the agenda. Picn et al. [24] proposed to combine hyperspectral and spatial characteristics of materials to form feature vectors to identify non-ferrous metals. Chen et al. [25] used transfer learning to classify data of small samples of non-ferrous metals on the basis of traditional image recognition techniques. Han et al. [26] also proposed a scrap classification method based on computer image recognition.

The above scrap classification methods use deep learning methods to classify scrap, or based on machine vision on the physical and chemical properties of materials, it can be predicted that deep learning has great prospects in the application of scrap classification.

2.2. Personalized Federated Learning

FL aims to use the local data distributed in each terminal device to jointly train a unified model, and upload model parameters instead of uploading local data to protect user data privacy [7,27,28]. However, in steel scrap classification scenarios, different environments and different user characteristics need to be considered, resulting in data with Non-IID characteristics in distribution, which seriously affects the accuracy of steel scrap classification in FL.

For Non-IID problems, FL has already developed a personalized method to solve them. Existing solutions are extensively discussed in [29] by Tan et al. Li et al. [30] proposed FedProx. On the basis of minimizing the global experience loss, the ℓ_2 -norm constraint is introduced to the loss function of the local training, so that the local update should not be too far away from the initial global model. Sai et al. [31] proposed a new algorithm (SCAFFOLD), which uses control variates (variance reduction) to correct for the “client-drift” in its local updates, and could taken advantage of similarity in the client’s data yielding even faster convergence. In [13], Wang et al. use the parameters in the local training process as the cognitive basis and calculate Earth mover’s distance to quantify the differences between different models. Presotto et al. [32] proposed a federated clustering algorithm FedCLAR, which grouped clients based on the similarity of client models, so as to better identify and distinguish client data with different distributions. Yan et al. [12] proposed ICFL, which can dynamically determine the cluster structure of clients during

each training round and aggregate a personalized model for each cluster. Long et al. [15] proposed multicenter FL, which forms multiple personalized models through clustering based on distributed similarity of data and optimizes them individually.

We focused on the clustering-based personalized FL, and found that the existing clustering methods rely on artificial preset cluster number, there is a lack of cooperation between clients of different clusters, and the model aggregation of clients within clusters does not consider the correlation between data distributions in a fine-grained manner. This inspired us to develop an end-to-end self-clustering approach to cluster a personalized model for each client that is most relevant to them.

3. Problem Statement

3.1. Federated Learning

There are N clients, and each client has its own local scrap steel data $D_i (i \in [1, N])$ and the loss on j th sample $(x_{i,j}, y_{i,j})$ is $l(x_{i,j})$. Moreover, the total loss of the client on its local data is $F_i(w) = \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} l(x_{i,j})$, where w is the model parameter, and $|D_i|$ is the length of the dataset. The overall loss after aggregation by the FL server is:

$$F(w) = \frac{\sum_{i=1}^N |D_i| F_i(w)}{|D|}, \tag{1}$$

where $|D| = \sum_{i=1}^N |D_i|$ is the total size of all client datasets.

Suppose $F^* = \sum_{i=1}^N \alpha_i F_i(w^*)$ is the smallest overall loss, where w^* is the optimal parameter that causes the model to converge and α_i is the learning rate of gradient descent algorithm. Let $\Delta = |F^* - \sum_{i=1}^N \alpha_i F_i^*|$, F_i^* be the optimal loss, then:

$$\Delta = \left| \sum_{i=1}^N \alpha_i F_i(w^*) - \sum_{i=1}^N \alpha_i F_i^* \right|, \tag{2}$$

where Δ is the difference between the optimal model of the client and the optimal model aggregated by the server, and its value can reflect the deviation of the aggregate model in the global data distribution.

Generally, the value of α_i is the proportion of each client to the total data. In this case, if the distribution of the clients' data is Non-IID, then $F_i(w^*) \neq F_i^*$ and $\Delta \neq 0$, resulting in the global model of server aggregation deviating from the global distribution.

In summary, the optimization objective of scrap steel classification based on FL can be summarized as follows:

$$w^* = \arg \min_w F(w) \tag{3}$$

$$\begin{aligned} \text{s.t. } & D_i \sim X(i), X(i) \neq X(j), \\ & i \neq j, \\ & \Delta < \delta, \end{aligned} \tag{4}$$

where $X(\cdot)$ represents the distribution of the data and δ represents an upper bound on the difference between the loss at the convergence point and the optimal loss.

3.2. Personalized FL-Based Scrap Steel Classification

In the Non-IID scenario, the global model w^* obtained by the federated average algorithm may be a local optimal minimum point for the client, which cannot meet the optimization requirements of each client [10,33]. Existing work proposes personalized FL, where customers optimize their local goals while participating in server collaborative training:

$$w_1^*, \dots, w_N^* = \arg \min \Phi(F(w_1), \dots, F(w_N)), \tag{5}$$

where $\Phi(\cdot)$ is set to $\sum_{i=1}^N a_i \cdot F(w_i)$ and a_i is the aggregation weights for w_i .

On the basis of PFL, this paper proposes federated steel scrap classification based on self-attention hierarchical aggregation. The optimization goal can be summarized as follows:

$$w_1^*, \dots, w_N^* = \arg \min \Phi(w_1, \dots, w_N) \quad (6)$$

$$s.t. \quad D_i \sim X(i), X(i) \neq X(j), i \neq j.$$

where D_i is the local dataset of client i , with different distributions ($X(i) \neq X(j)$), and $\Phi(\cdot)$ is a different aggregation function for different clients' models, as calculated by:

$$\Phi(w_i) = \sum_{j=1}^N \alpha_{i,j} \cdot w_j \quad (7)$$

$$= \sum_{j=1}^N \frac{\|w_i - w_j\|_2}{\sum_{j=1}^N \|w_i - w_j\|_2} \cdot w_j,$$

where $\alpha_{i,j}$ is the aggregation weight between w_i and w_j and $\|w_i - w_j\|_2$ is the L2-norm between w_i and w_j , which signifies their correlation. Through this method of aggregation, similar clients are allocated greater aggregate weights, enabling the personalized model aggregated for the clients to benefit more from those that are more closely related to themselves.

4. Overview and Implementation

4.1. Overview

The overall process of FedScrap is illustrated in Figure 1, which consists of two main parts: the model training on the client side and the model aggregation on the server side, as detailed below.

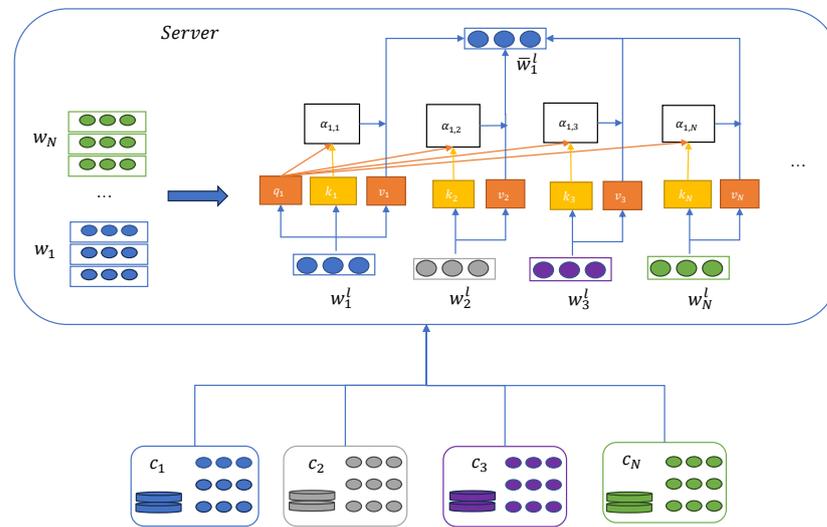


Figure 1. Overview of FedScrap. The dots in different colors represent different client's Scrap steel classification models, which are trained locally using pre-processed waste steel data. The server model aggregation is achieved by layer-by-layer aggregating a personalized model relevant to each client through the self-attention mechanism.

1. Preprocessing of scrap data and local training of classification models. Due to the complex scrap detection environment, which contains a large number of stacked scrap and background information, clients need to preprocess its scrap data and extract scrap features conducive to classification. Then, clients build their own neural network models with the same structure (such as resnet18, etc.) for training on the processed data.

2. Personalized aggregation of parameters based on self-attention. The server receives the model parameters trained by the clients and aggregates the personalized model for each client using self-attention. The core of self-attention is to measure the model similarity between clients, which can represent the distributed similarity between clients' data, and then assign greater aggregate weight to those clients' models that are more relevant to them when aggregating models for clients.

4.2. Implementation and Algorithm Description

4.2.1. Preprocessing of Scrap Data and Local Training of Classification Models

The scrap recycling sites collect the image data of scrap steel, and decomposes the scrap steel data in JPG format into RGB three primary color pixels through the image data processing function in python. The converted scrap data are unified into input data in the same format by pre-processing operations, such as cutting, de-noising, and normalization. The processed data are called the client's local data D_i , where D is the abbreviation for the dataset and client i is the i -th scrap recycling sites.

After processing the data, the clients locally build a deep learning model with the same structure to learn the features of the data. Different types of models have different feature extraction capabilities, but generally have input layers that are in contact with the data, fully connected layers for decision making, and a backbone network to extract features. The backbone network adopted in this paper is the residual network ResNet structure, which has the advantage of extracting the original features of the data and preventing the gradient disappearance caused by data heterogeneity. For ease of description, the overall model of the i -th client is denoted as w_i .

In order to speed up the training of the model, the scrap data are divided into small batch data and then iteratively fed into the network model for training. The training method adopts stochastic gradient descent (SGD) algorithm, which is a fast iterative training algorithm for small batch data. SGD calculates the error loss $F(w_i)$ between the sample label estimated by the network model and the real label through forward propagation, and inversely solves the gradient of the model parameters, i.e., $\nabla F(w_i)$, to update the parameters so as to reduce the loss. In general, the client's scrap classification model parameter w_i was updated in round t as follows:

$$w_i^t = w_i^{t-1} + \eta_i \nabla F(w_i^t), \quad (8)$$

where $\nabla F(w_i^t)$ and η_i represent the model gradient and the step size of gradient descent, respectively. After local training, we upload the parameters or gradient values of the steel scrap classification model to the server for model aggregation.

4.2.2. Personalized Model Aggregation Based on Self-Attention

The server receives the model parameters uploaded by the clients and aggregates them into a global shared unified model according to the aggregation strategy, as follows:

$$w = \sum_{i=1}^N \alpha_i w_i, \quad (9)$$

where α_i is the aggregation weight. The traditional aggregation strategy is to weight the model according to the data volume of the clients, i.e., $\alpha_i = |D_i| / \sum_{i=1}^N |D_i|$, but this approach involves the sensitive information of the data volume $|D_i|$, and does not take into account the model bias caused by the data distribution.

In this paper, we propose a personalized model aggregation strategy based on layer-wise self-clustering, which aims to customize a most relevant model for each client to adapt to its data distribution. According to representation learning, the features presented by two models with the same structure after training with different data show decreasing similarity at each layer of the model. In order to examine the fine-grained features between clients, we quantified the layer-wised correlation of model parameters between clients, which

is calculated by $d_{i,j}^l = \|w_i^l - w_j^l\|_2$. That is, $d_{i,j}^l$ reflects the degree of similarity between the layers of the model between the two clients, which is determined by the internal relationship of the data distribution between the clients.

On this basis, we aggregate a personalized model for each client individually. Specifically, the server takes each client as a cluster center to aggregate the models of other clients, and the weights of the aggregation are dynamically assigned according to the similarity of the parameters between them, that is:

$$\alpha_{i,j}^l = \frac{d_{i,j}^l}{\sum_{j \in N} d_{i,j}^l}, \quad (10)$$

$$\sum_{j=1}^N \alpha_{i,j}^l = 1. \quad (11)$$

In order to improve the ability of the aggregation model to process heterogeneous data, a larger aggregation weight $\alpha_{i,j}^l$ is assigned between clients with similar data, which makes the i -th client more “concerned” about the model parameters of the j -th client at layer j . In other words, there is more cooperation between similar clients, which is mutually desirable and satisfies each other’s needs.

Then, the server repeats the aggregate weight $\alpha_{i,j}^l$ between the two clients and customizes the aggregate model for each client based on the weight, that is:

$$w_i^l = \sum_{j \in N} \alpha_{i,j}^l w_j^l, \quad (12)$$

where w_i^l is the model parameter of layer l aggregated by the server for client i . In a Non-IID environment, calculating the model similarity between clients layer by layer can facilitate effective cooperation among multiple parties, but a significant problem with this is the complexity of the calculations, which require frequent and repeated access to each client’s parameters at each layer of the model. Therefore, it is necessary to develop a model aggregation method capable of parallel computation.

In this paper, the idea of self-attention is adopted, which can calculate the attention weight of serialized data in parallel, so as to extract the correlation within the sequence data. In this way, the more relevant content of each client is deeply extracted into the server model, reducing the aggregation of personalized features while preserving shared features. Specifically, the personalized federated scrap detection process based on self-attention is as follows:

1. Serialize the model parameters for each client of the same layer into an input vector, denoted as $w^l = [w_1^l, w_2^l, \dots, w_N^l]$ for any $l \in L$.
2. Multiply each vector w by three coefficients a^q , a^k , and a^v to get three vectors: query, key, and value, that is, $Q = A^q \cdot w^l$, $K = A^k \cdot w^l$ and $V = A^v \cdot w^l$, where $A = [a_1, a_2, \dots, a_N]$.
3. The attention score (similarity score) is calculated for the clients by matrix multiplication, i.e., $\alpha^l = Q \cdot K$, where $\alpha_{i,j}^l = a^q \cdot w_i^l \cdot a^k \cdot w_j^l$, which reflects the degree of similarity between the parameters.
4. The attention scores were normalized using softmax and other methods, i.e., $\hat{\alpha}^l = \text{softmax}(\alpha^l)$.
5. Finally, the aggregated parameters are obtained by multiplying the normalized attention score by the parameter vector, i.e., $\hat{w}^l = \hat{\alpha}^l \cdot w^l$, where $\hat{w}_i^l = \sum_{j=1}^N \hat{\alpha}_{i,j}^l \cdot w_j^l$.

The aggregation algorithm of federated scrap detection is shown in Algorithm 1.

Algorithm 1 Personalized model aggregation based on self-attention

-
- 1: **for** $i \in N$ **do**
 - 2: The client local model training: $w_i^t = w_i^{t-1} + \eta_i \nabla F(w_i^t)$.
 - 3: The client uploads its local model parameter w_i^t .
 - 4: **end for**
 - 5: **for** $l \in L$ **do**
 - 6: The server serializes the model parameters of the clients: $w^l = [w_1^l, w_2^l, \dots, w_N^l]$.
 - 7: Parameter vector coefficients: $Q = A^q \cdot w^l$, $K = A^k \cdot w^l$ and $V = A^v \cdot w^l$.
 - 8: The server calculates the attention score for the clients: $\alpha^l = Q \cdot K$.
 - 9: Attention score normalization: $\hat{\alpha}^l = \text{softmax}(\alpha^l)$.
 - 10: Aggregate model parameters according to attention scores: $\hat{w}_i^l = \sum_{j=1}^N \hat{\alpha}_{i,j}^l \cdot w_j^l$.
 - 11: **end for**
 - 12: The server sends the aggregated model parameters \hat{w}_i to all clients.
-

5. Evaluation**5.1. Set Up**

In order to effectively evaluate the performance of the method proposed in this paper, we conduct experimental verification on a self-built FL framework with ten clients. The framework is deployed on a server with 56 NVIDIA GeForce RTX 3090 Ti graphic cards.

Since there is no open source scrap detection dataset at present, we collected a batch of scrap image data from a steel mill in China, which has six common types of scrap, including silicon steel sheet, rebar, steel slag hot pressed block, heavy scrap, square pressing block, and messy scrap, as shown in the Figure 2, and the total sample size was about 1000 images.



Figure 2. Examples of different scrap types in self-built datasets.

In terms of the depth algorithm for extracting data features, we use three common models for verification, and the three model pairs are shown in Table 1.

Table 1. Models used to extract data features.

Models	Mechanism	Parameters	Layers	Activation Function
LeNet-5	Convolutional Layers	60 k	7	ReLU
ResNet-18	Convolutional Layers	11.7 M	18	ReLU
ViT-12	Self-Attention	86.57 M	12	ReLU/Layer Normalization

In order to compare the performance of the proposed layer-based method in terms of model prediction accuracy, client's local model accuracy, etc., the comparison method in this paper is as follows:

1. FedAvg [7], an FL classic algorithm that collects and averages model parameters across clients.
2. FedProx [30], an FL method that restricts the client's update direction to enhance the performance of the global model, has a hyperparameter μ for the constraint we set to 0.01.
3. ICFL [12], a clustering FL algorithm that automatically clusters clients and aggregates clustering models according to the correlation between clients without setting the number of clusters.
4. CPFL [13], a clustering FL algorithm that individually aggregates the personalized models associated with them for each client.

The comparison methods use the same dataset, network model, and hyperparameter settings.

5.2. Overall Accuracy Comparison

We divide the self-built dataset into ten clients using the Dilliclet distribution, with a Dilliclet coefficient α of 0.01, and each client has a different number or category of scrap data. Then, resnet and Vit models are used to train and test the accuracy of several comparison methods on each client. First, we averaged the accuracy of the client to observe the global average accuracy of the whole and evaluate the performance of each method. The result records are shown in Table 2, where \pm represents the standard deviation between the accuracy of the clients.

Table 2. Overall accuracy comparison of methods on different models (the unit of accuracy is %).

Methods Models	Local	FedAvg	FedProx	CPFL	ICFL	FedScrap
ResNet-18	77.868 \pm 14.88	97.655 \pm 6.3	98.834 \pm 2.06	99.583 \pm 1.32	99.322 \pm 1.43	99.655 \pm 1.09
VIT	89.857 \pm 15.7	98.199 \pm 3.57	94.735 \pm 10.38	97.776 \pm 3.82	97.164 \pm 3.35	98.985 \pm 1.74
LeNet	80.593 \pm 18.89	98.797 \pm 10.99	95.5 \pm 9.78	89.819 \pm 15.6	94.468 \pm 7.24	97.899 \pm 2.4

The results indicate that under the Non-IID data distribution, the unprocessed local scrap steel classification model exhibits extremely low accuracy performance, and there is a substantial variance in accuracy across clients. This suggests that while some clients achieve high accuracy, others remain at very low levels, which is clearly unsatisfactory. Traditional FL methods such as FedAvg and FedProx can enhance classification accuracy; however, their performance is highly unstable across different models, exhibiting considerable variance. This is primarily due to their reliance on a single shared model maintained on the server to address the data from all clients. Cluster-based PFL methods significantly mitigate this issue, achieving a high average accuracy with reduced variance. Our proposed FedScrap outperforms these methods, demonstrating high robustness across various models. Both our accuracy and variance are superior to those of the comparative methods, which is attributed to the personalized models tailored for each client, absorbing the most relevant knowledge for their respective contexts.

For a more intuitive comparison, we plotted the test results of different models, as shown in Figure 3.

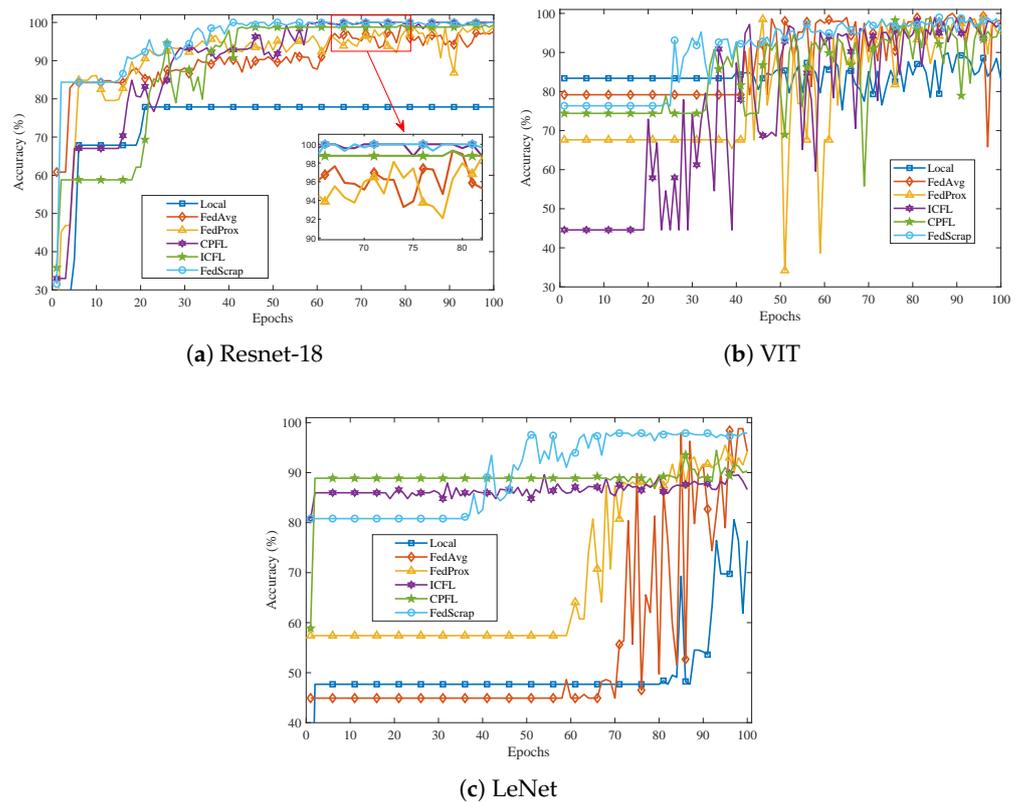


Figure 3. Accuracy comparison of methods on different models.

The analysis reveals that in scenarios characterized by an imbalance in scrap types, the classification model trained exclusively on local data exhibits low overall accuracy. This shortcoming arises because certain client-specific samples are too limited to yield effective training outcomes. Consequently, there is a pressing need for FL to collaborate with these dispersed clients and enhance the model’s performance. The yellow curve, depicting conventional FL, demonstrates a notable enhancement in model accuracy. However, the accuracy fluctuates markedly and convergence is achieved gradually. This volatility is attributed to the data heterogeneity resulting from the unbalanced distribution of scrap types, which challenges a single global model’s capacity to cater to the diverse needs of all clients.

Although existing cluster-based personalized methods can marginally elevate model accuracy, they are accompanied by significant fluctuations in the early stages, suggesting a slow convergence rate. This delay is rooted in the initial step of identifying cluster centers, followed by the aggregation of personalized models around these centers. The choice of cluster centers is critical and can significantly impact the model’s convergence. In contrast to these approaches, our proposed FedScrap method treats each client as a clustering center, effectively positioning them as individual servers. This strategy facilitates the aggregation of highly relevant personalized models that have absorbed sufficient bespoke knowledge from similar clients, thereby aligning closely with their respective data distributions. As a result, FedScrap not only enhances model precision, but also ensures a smooth convergence process.

5.3. Comparison of Accuracy Differences between Methods among Clients

To more finely compare the individual variations of each method at the client level and to identify the reasons why the accuracy variances of other methods are not sufficiently impressive, we have recorded and plotted the accuracy box plots for each client. The results are shown in Figure 4, where the three box plots represent the three backbone network models used.

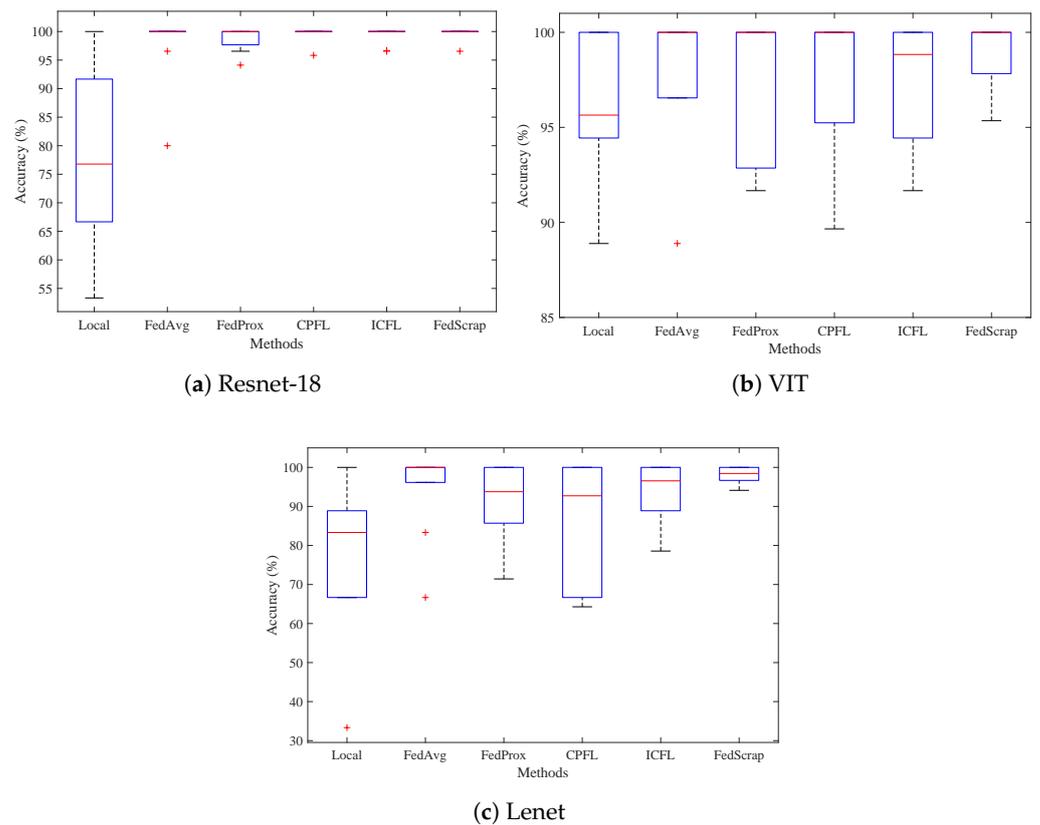


Figure 4. Accuracy comparison of different methods on boxplots.

It can be observed that the three graphs depict a similar pattern of performance: local performs the worst, with an accuracy of merely 50%, while FedScrap exhibits the best performance, consistently maintaining accuracy above 90%, and the other methods each have their strengths but also suffer from notable drawbacks.

Due to the impact of data Non-IID, there are certain differences in the types of scrap metal among clients. Some clients have extremely limited samples for certain types of scrap metal, resulting in poor accuracy, while those with better data resources perform well. Although FedAvg and FedProx, as representatives of a single global model, have combined client training, the differentiated data lead to poor performance on some clients. The cluster-based methods CPFL and ICFL group clients based on their similarity, maintaining a shared model within each group. However, this breaks the close connection between clients, leading to a decline in average accuracy.

Unlike these methods, FedScrap not only fully considers the relevance of data between clients but also uses this relevance to strengthen the connection between clients. As a result, the personalized models tailored for each client can absorb useful information from other clients based on their own data distribution. These aspects contribute to FedScrap's average accuracy among clients reaching up to 97%, with a standard deviation of only about 2%.

6. Conclusions

This paper introduced FedScrap, a layer-wised personalized FL framework for scrap detection. Utilizing the self-attention mechanism, FedScrap coordinates distributed scrap data to train a robust scrap classification model. The framework addressed the challenge of non-independent co-distribution in distributed scrap data by employing the self-attention concept to aggregate a personalized model for each client that is most relevant to its specific data. We also collected scrap pictures from a steel mill and labeled them to make scrap classification dataset, and carried out verification experiment on this dataset. Experimental results show that FedScrap accurately classifies distributed scrap data with an impressive accuracy rate of 90%.

Author Contributions: Conceptualization, W.Z. and L.W.; methodology, W.Z.; software, L.W.; validation, D.D. and L.W.; formal analysis, D.D.; investigation, L.W.; resources, L.W.; data curation, W.Z.; writing—original draft preparation, W.Z.; writing—review and editing, W.Z. and D.D.; visualization, D.D.; supervision, L.W.; project administration, W.Z.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China, grant number 62172003.

Data Availability Statement: The data can be shared up on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xu, W.; Xiao, P.; Zhu, L.; Zhang, Y.; Chang, J.; Zhu, R.; Xu, Y. Classification and rating of steel scrap using deep learning. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106241. [[CrossRef](#)]
2. Gao, Z.; Lu, H.; Lei, J.; Zhao, J.; Guo, H.; Shi, C.; Zhang, Y. An RGB-D-Based Thickness Feature Descriptor and Its Application on Scrap Steel Grading. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5031414. [[CrossRef](#)]
3. Zhang, R.; Xiong, K.; Lu, Y.; Fan, P.; Ng, D.W.K.; Letaief, K.B. Energy efficiency maximization in RIS-assisted SWIPT networks with RSMA: A PPO-based approach. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1413–1430. [[CrossRef](#)]
4. Zhang, T.; Xu, C.; Lian, Y.; Tian, H.; Kang, J.; Kuang, X.; Niyato, D. When Moving Target Defense Meets Attack Prediction in Digital Twins: A Convolutional and Hierarchical Reinforcement Learning Approach. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3293–3305. [[CrossRef](#)]
5. Zhang, T.; Xu, C.; Shen, J.; Kuang, X.; Grieco, L.A. How to Disturb Network Reconnaissance: A Moving Target Defense Approach Based on Deep Reinforcement Learning. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 5735–5748. [[CrossRef](#)]
6. Zhang, T.; Xu, C.; Zou, P.; Tian, H.; Kuang, X.; Yang, S.; Zhong, L.; Niyato, D. How to mitigate DDOS intelligently in SD-IOV: A moving target defense approach. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1097–1106. [[CrossRef](#)]
7. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
8. Zhang, R.; Xiong, K.; Lu, Y.; Gao, B.; Fan, P.; Letaief, K.B. Joint coordinated beamforming and power splitting ratio optimization in MU-MISO SWIPT-enabled HetNets: A multi-agent DDQN-based approach. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 677–693. [[CrossRef](#)]
9. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-IID data. *arXiv* **2018**, arXiv:1806.00582.
10. Xu, J.; Tong, X.; Huang, S.L. Personalized Federated Learning with Feature Alignment and Classifier Collaboration. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2022.
11. Cheng, D.; Zhang, L.; Bu, C.; Wang, X.; Wu, H.; Song, A. ProtoHAR: Prototype Guided Personalized Federated Learning for Human Activity Recognition. *IEEE J. Biomed. Health Inform.* **2023**, *24*, 3900–3911. [[CrossRef](#)]
12. Yan, Y.; Tong, X.; Wang, S. Clustered Federated Learning in Heterogeneous Environment. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–14. [[CrossRef](#)]
13. Wang, J.; Xu, G.; Lei, W.; Gong, L.; Zheng, X.; Liu, S. Cpfl: An effective secure cognitive personalized federated learning mechanism for industry 4.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7186–7195. [[CrossRef](#)]
14. Li, B.; Chen, S.; Yu, K. FeDDkw–Federated Learning with Dynamic Kullback–Leibler-divergence Weight. In *ACM Transactions on Asian and Low-Resource Language Information Processing*; Association for Computing Machinery: New York, NY, USA, 2023.
15. Long, G.; Xie, M.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J. Multi-center federated learning: Clients clustering for better personalization. *World Wide Web* **2023**, *26*, 481–500. [[CrossRef](#)]
16. Ghosh, A.; Chung, J.; Yin, D.; Ramchandran, K. An efficient framework for clustered federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19586–19597. [[CrossRef](#)]
17. Duan, M.; Liu, D.; Ji, X.; Wu, Y.; Liang, L.; Chen, X.; Tan, Y.; Ren, A. Flexible clustered federated learning for client-level data distribution shift. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 2661–2674. [[CrossRef](#)]
18. Vahidian, S.; Morafah, M.; Wang, W.; Kungurtsev, V.; Chen, C.; Shah, M.; Lin, B. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 10043–10052.
19. Xu, B.; Xia, W.; Zhao, H.; Zhu, Y.; Sun, X.; Quek, T.Q. Clustered federated learning in Internet of Things: Convergence analysis and resource optimization. *IEEE Internet Things J.* **2023**, *11*, 3217–3232. [[CrossRef](#)]
20. Tu, Q.; Li, D.; Xie, Q.; Dai, L.; Wang, J. Automated Scrap Steel Grading via a Hierarchical Learning-Based Framework. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5022313. [[CrossRef](#)]
21. Smirnov, N.V.; Trifonov, A.S. Deep Learning Methods for Solving Scrap Metal Classification Task. In Proceedings of the 2021 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 5–11 September 2021; pp. 221–225.
22. Williams, K.C.; O’Toole, M.D.; Peyton, A.J. Scrap metal classification using magnetic induction spectroscopy and machine vision. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2520211. [[CrossRef](#)]

23. Diaz-Romero, D.J.; Van den Eynde, S.; Sterkens, W.; Engelen, B.; Zaplana, I.; Dewulf, W.; Goedemé, T.; Peeters, J. Simultaneous mass estimation and class classification of scrap metals using deep learning. *Resour. Conserv. Recycl.* **2022**, *181*, 106272. [[CrossRef](#)]
24. Picón, A.; Ghita, O.; Whelan, P.F.; Iriondo, P.M. Fuzzy Spectral and Spatial Feature Integration for Classification of Nonferrous Materials in Hyperspectral Data. *IEEE Trans. Ind. Inform.* **2009**, *5*, 483–494. [[CrossRef](#)]
25. Chen, S.; Hu, Z.; Wang, C.; Pang, Q.; Hua, L. Research on the process of small sample non-ferrous metal recognition and separation based on deep learning. *Waste Manag.* **2021**, *126*, 266–273. [[CrossRef](#)]
26. Han, S.D.; Huang, B.; Ding, S.; Song, C.; Feng, S.W.; Xu, M.; Lin, H.; Zou, Q.; Boularias, A.; Yu, J. Toward fully automated metal recycling using computer vision and non-prehensile manipulation. In Proceedings of the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, France, 23–27 August 2021; pp. 891–898.
27. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [[CrossRef](#)]
28. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [[CrossRef](#)]
29. Tan, A.Z.; Yu, H.; Cui, L.; Yang, Q. Towards personalized federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 9587–9603. [[CrossRef](#)] [[PubMed](#)]
30. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
31. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. Scaffold: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; pp. 5132–5143.
32. Presotto, R.; Civitarese, G.; Bettini, C. Fedclar: Federated clustering for personalized sensor-based human activity recognition. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), Biarritz, France, 11–15 March 2022; pp. 227–236.
33. Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; Guan, H. FedALA: Adaptive local aggregation for personalized federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 11237–11244.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.