
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
 Volume 25, Issues 6, pp. 4576–4588, DOI 10.12694/scpe.v25i6.3305

FEDERATED DIVERSE SELF-ENSEMBLING LEARNING APPROACH FOR DATA HETEROGENITY IN DRIVE VISION

M.MANIMARAN *AND V.DHILIPKUMAR †

Abstract. Federated learning (FL) has developed as an efficient framework that can be used to train models across isolated data sources while also protecting the privacy of the data. In FL a common method is to construct local and global models together, with the global model (server) informing the local models and the local models (clients) updating the global model. Most present works assume clients have labeled datasets and the server has no data for supervised learning (SL) problems. In reality, clients lack the competence and drive to identify their data, while the server may host a tiny amount. How to reasonably use server-labeled and client-unlabeled data is crucial in semi-supervised learning (SSL) and Cclientdata heterogeneity is widespread in FL. However, inadequate high-quality labels and non-IID client data, especially in autonomous driving, decrease model performance across domains and interact negatively. To solve this Semi-Supervised Federated Learning (SSFL) problem, we come up with a new FL algorithm called FedDSL in this work. We use self-ensemble learning and complementary negative learning in our method to make clients' unsupervised learning on unlabeled data more accurate and efficient. It also coordinates the model training on both the server side and the clients' side. In an important distinction to earlier research that kept some subset of labels at each client, our method is the first to implement SSFL for clients with 0% labeled non-IID data. Our contributions include the effectiveness of self-ensemble learning by using confidence score vector for calculating only for the current model performing data filtering and initiated negative learning by showing the data filtering performance in the beginning rounds. Our approach has been rigorously validated on two significant autonomous driving datasets, BDD100K and Cityscapes, proving to be highly effective. We have achieved state-of-the-art results and the metric that is utilized to evaluate the effectiveness of each detection task is mean average precision (mAP@0.5). Astonishingly FedDSL performs nearly as well as fully-supervised centralized training approaches, despite the fact that it only uses 25% of the labels in the Global model.

Key words: Federated Learning, Data Heterogeneity, Autonomous Vehicle, Ensemble Learning.

1. Introduction. Machine Learning (ML)-based services and apps are becoming more and more important because of the need to protect data safety and security. It can be hard for ML providers to gather and manage data, follow General Data Protection Regulation (GDPR) rules, and keep personal data from getting lost, misused, or abused [1]. If you want to do global collaborative learning without sharing the original training data, Federated Learning (FL) can help. But it's not totally secret since the model parameters include private information. FL is a way to learn that a single server manages a group of devices used by many people. The server delivers the most recent model to clients at every training epoch. After that, the clients modify the model using their private data sets. [2]. The server takes all of these changes and applies them to the core model. FL lets you make a whole ML model without disclosing client data or using their computing power, which makes the server load less. The main elements of the FL concept are as follows:

* Decentralized Training: Gathering all data in one location and by training the model directly on individual devices, FL ensures that data privacy is preserved.
* Updates: Each device updates its model parameters and sends them to a central server after undergoing model training with its own data.
* Global Aggregation: The server collects the updated parameters from all devices. Utilizes them to train a model.
* Model Updates: The improved global model is subsequently sent back to the devices so they can train it using their data.

*Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai, India. (*drmanimaranm@gmail.com)

†Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai, India. (*vdhilipkumar@veltech.edu.in)

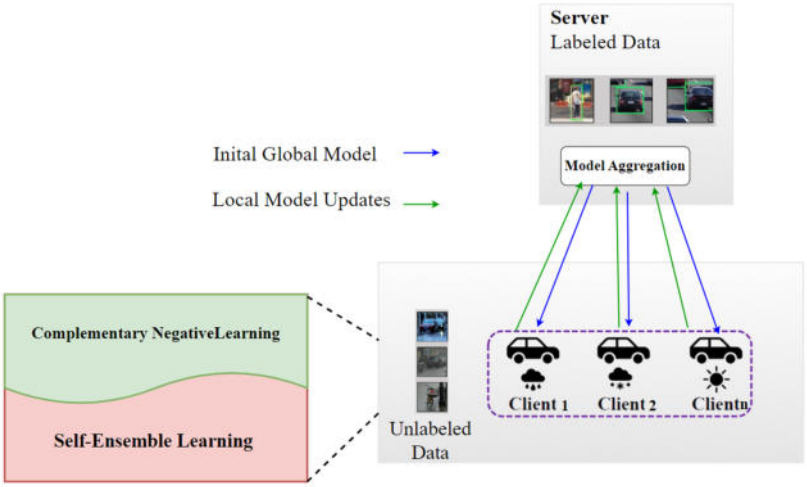

Fig. 1.1: An Overview of FedDSL

In FL applications like keyboard prediction, labelling data isn't always needed, but it can be pricey to get big labelled datasets. Currently, most of the research is focused on building ML models from unlabeled data. However, not many of the current studies try to improve FL [3], which leaves a good area for future research. FL methods mostly work with supervised learning, which means that all of the data is labelled. But getting fully labelled data can be hard because it is possible that the participants will lack knowledge or interest in the subject [4]. This makes the problem of FL out of full labels very important in real-world applications like autonomous driving.

In SemiFL [5], clients have data that hasn't been labelled at all and can train over multiple local epochs to cut down on connection costs. The server, in contrast, stores some tagged data. However, this system may become less useful if labelled and unlabeled data are separated, as it becomes more difficult to apply these methods to FL. While customers only have access to unlabeled data, label-at-server scenarios are more problematic due to the fact that only the server possesses labelled data. To overcome the information gap between labelled and unlabeled data in autonomous driving, a new strategy is needed where there is no direct exchange of data.

With semi-supervised federated learning (SSFL) [6], clients' unsupervised learning on unnamed data is made more accurate and efficient by using self-ensemble learning and complementary negative learning. On the client side as well as the server side, SSFL is responsible for managing the model training. Limited high-quality IDs and non-IID client data can be a challenge, especially in situations like self-driving cars. As a solution to these problems, the Semi-Supervised Federated Object Detection (SSFOD) [7] system was designed for cases where clients have unlabeled data and the server only has labelled data. The fact that SSFOD has never been used before for clients with zero percent labelled non-IID data is interesting. This is a significant departure from earlier studies that attached labels to each patient.

There have been the following problems with the aforementioned research:

* primarily using datasets like CIFAR, Fashion MNIST, ImageNet and COCO can be challenging because to their small size and complexity, but SSFOD poses far larger challenges.
* Following that, we proceeded to investigate difficult FL cases that involved clients who had unlabeled data from a various category of object than the labelled data that was stored on the server. For example, when the weather changes, real-life FL scenarios aren't always identical, so we made this adjustment. Because client records are so diverse, this approach accounts for that fact.

For the purpose of overcoming these issues of SSFOD that still need to be taken into consideration, we introduce FedDSL (Federated Diverse Self-Ensemble Learning) a personalized multi-stage training technique for the

SSFOD Framework as shown in Figure 1.1.

We proposed a FedDSL and Negative Learning for a consistency-based regularization scheme and a cost-based pseudo-label mining algorithm for the SSFOD framework making the predictors more resilient to biased representations caused by local data heterogeneity.

Our recently added features offer a novel approach to using FL unlabeled data to improve non-IID identification performance, particularly for dynamic objects, which has been ignored in previous research. We assert that our methodology and tests set a useful standard for future research in many domains, despite the fact that SSFOD has received very little attention in the scientific community.

2. Related Works.

2.1. FL: Advancements and Challenges. Recently, FL has caught the attention of the public as a method that protects users' privacy while simultaneously maximising the potential of distributed data. Despite the fact that FL has come a long way, its adaptability and applicability to other real-world problems may be limited. This is due to the fact that the majority of studies have concentrated on categories of tasks. In the future, advanced FL approaches will pave the way for collaborative learning from dispersed data sources, which will be of great benefit to a number of different industries, including autonomous driving, healthcare, and finance, among others. Future vehicular IoT systems [12], With growing privacy concerns, FL will be crucial for intelligent transport systems and cooperative autonomous driving to optimise the use of scarce communication, computation, and storage resources. This publication delves into the latest advancements in smart healthcare virtual reality [13], including resource-aware, secure, incentive, and personalized FL designs. Health data management, COVID-19 detection, medical imaging, and remote monitoring are just a few of the important healthcare domains covered in this analysis. It also examines current FL-based projects and highlights important lessons learned.

In FL, addressing data heterogeneity is crucial since clients often have data with different distributions, which can affect the global model's performance. Adaptive aggregation algorithms and local model fine-tuning are two of the many approaches suggested by researchers to deal with non-IID data and meet this difficulty. This paper [14] examines the convergence of federated versions of adaptive optimizers such as Yogi, Adam, and Adagrad in non-convex scenarios with heterogeneous data. The effects of client heterogeneity on the efficacy of communication are shown by the results. A generalization of FedAvg, FedProx [15] is a framework that deals with federated network heterogeneity. It handles statistical and system heterogeneity by ensuring convergence for learning over non-identical distributions and enabling devices to execute varied tasks. For client-specific model adaptation, researchers investigate techniques such as model distillation and meta-learning. Quantization, sparsification, and a supernet are solutions that decrease overhead while retaining model performance, which is critical in FL for efficient communication.

2.2. YOLO based Semi Supervied Object Detection Technique . Additionally, a significant amount of effort has been put into Semi Supervised Object Detection (SSOD), the primary goal of which is to improve detection performance in general by improving pseudo labels. The primary goal of developing SSL (Semi-Supervised Learning) methods for object detection is to produce consistent and trustworthy pseudo labels using pretrained architectures and robust data augmentation strategies. Two-stage detectors usually outperform their one-stage counterparts, and single-stage detectors, such as YOLO, have a hard time with SSL techniques. Researchers are coming up with innovative solutions to fix the problem of existing SSL approaches failing with single-stage detectors.

Epoch Adaptor, Dense Detector, and Pseudo Label Assigner make up the Efficient Teacher Framework [16], a one-stage SSOD training system. Implementing a baseline model, a pseudo designation process, and Dense Detector into the framework improves its performance. Dense Detector is able to acquire globally consistent features and the system prevents bias caused by poor pseudo labels; this makes training independent of the proportion of labelled data. Author [17] introduced a YOLO based method through the incorporation of domain adaptation and the compact one-stage stronger detector YOLOv5, the performance of cross-domain detection will be improved. furthermore, to fix image-level discrepancies, use scene style transfer to cross-generate pseudo images across domains.

2.3. Semi-Supervised Federated Learning (SSFL). The SSFL method has recently surfaced as a potential solution to the issue of minimally labelled data in FL scenarios. Through the collaborative use of participant-owned labelled and unlabeled data, SSFL aims to enhance FL. Our research has focused on two main settings: labels-at-client and labels-at-server.

The two most critical cases in SSFL were defined according to the placement of the tagged data. In the first case, authors assume the common occurrence of clients having both labelled and unlabeled data. The second case involves a more challenging situation where the tagged data is exclusively accessible on the server. Our novel strategy to resolve the concerns voiced is known as FedMatch [18], short for Federated Matching. Compared to inexperienced hybrids of federated and semi-supervised learning approaches, FedMatch outperforms them in a number of ways, including a novel inter-client consistency loss and a dissection of the parameters for disjoint learning on labelled and unlabeled data.

Authors use the Semi-Supervised Federated Object Detection (SSFOD) framework when all the data is labelled on the server and there is no unlabeled data on the client. Thanks to FedSTO [7], a two-pronged approach, data transfer between servers and clients is handled securely. It uses orthogonality regularisation to improve visualisation divergence, selective training to prevent overfitting, and local Exponential Moving Average to assign high-quality pseudo labels (EMA).

2.4. Data Heterogeneity. When it comes to autonomous driving (AD), AutoFed is a FL platform that takes heterogeneity into account. Authors [24] goal is to make robust AD possible by making full use of multimodal sensory data on AVs. A client selection mechanism, an autoencoder-based data imputation method, and a novel model using pseudo-labelling are all part of the framework. The experimental results reveal that AutoFed outperforms the state-of-the-art methods in terms of precision and recall, and it also shows remarkable resilience in the face of bad weather. The Authors [25] trains an autonomous vehicle via federated learning. Data is gathered from an Udacity-based simulated vehicle over two tracks, and a Convolution Neural Network is used for training. The next step is to upload the modal to a server, where it will be integrated with other models to generate a new model. The car's performance is examined through multiple trainings, and the results show that the accuracy is iteration dependent and that merged models are less accurate.

3. Problem Statement. We focus on situations when 100% Unlabeled dataset in client side and 25% labelled dataset in server side for object identification task that involves a labeled dataset $D_l = [x_i^l, y_i^l]_{i=1}^{N_l}$ and unlabeled dataset $D_u = [x_i^u]_{i=1}^{N_u}$ where $N_u << N_l$. x^l as labeled images, y^l as annotations, it contains coordinates and differenet bounding boxes for objects. The dataset $\{x^l, y^l\}$ and the model parameterized by M^l are both stored on the server. Presumably, there are C clients, and they all have an unlabeled dataset $x^{u,C}$, Every client model, with parameters $M^{u,C}$ employs the identical architecture for detecting objects, represented by $f : (x, M) \rightarrow f(x, M)$, where each parameter M and input x are transformed into a collection of bounding boxes with associated class probabilities.

Heterogenity. Our research focuses on non-IID data that results from various weather conditions, such as cloudy, gloomy, rainy, and snowy. The three datasets—BDD100K, CityScapes, and SODA10M—display different class distributions and label densities. A framework for SSFOD that is capable of managing this diversity while preserving performance in a variety of contexts and with a variety of distributions is what we intend to create. When every client exhibits a well-balanced distribution of weather conditions, the data is considered to be a part of the independent and identically distributed (IID).

4. Proposed Methodology – FedDSL. To illustrate, FedDSL is an iterative algorithm that, similar to all FL algorithms, necessitates the exchange of local and global models between the server and the clients. Figure 4.1 illustrates this. As an alternative to conventional supervised FL, the server will use its labelled dataset to actively contribute to the global model's updates during training. This indicates that the server will be actively involved in the training process. Each FL round in FedDSL is comprised of four separate steps, which are as follows:

1. Initial Step: Server-side Supervised Learning
2. Second, have the clients download the global model.
3. Thirdly, client-side unsupervised learning
4. Finally: Submit local models to the Server

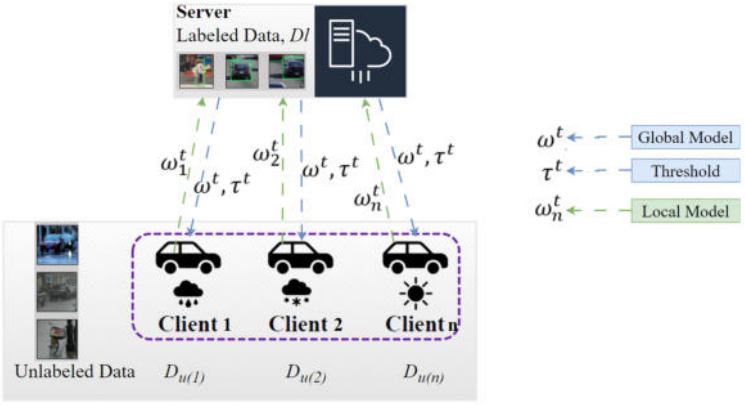

Fig. 4.1: Operation of FedDSL

4.1. Supervised Learning at Server. In this paper, author examines the image classification issue using an SSFL system, which consists of just one server and some clients.D_l is a labeled training dataset for server. We use x_i for the feature vector (image) and y_i for the label (image class) for each instance $i \in D_l$. Index of the class set $M = 1, .., m$ and hence, $y_i \in M$ for each instance i. Let us assume that the server possesses a limited labelled validation dataset of D_l^{Val} separated from the training dataset. D_u is unlabeled training dataset for individual client $M = 1, .., m$. The feature vector x_i of each instance $i \in D_u$ assume the dataset is unlabeled. At the clients, the total number of instances of unlabeled training data is greater than the number of instances of labelled training data that are stored on the server alone, which is denoted as $D_l << \sum_{n=1}^{N} D_u$. Our main objective is to successfully train a model f for classification, with ω serving as the parameter. probability distribution for predicted class is p(x,ω), x as input for the confidence score vector and model parameter of ω. p_m(x,ω) is the predicted class probability for the input x which belongs to class m under the model parameter ω, it is a confidence score. By simply using the formula f_m(x,ω)=arg$max_m p_m$(x;ω), A predicted hard label can be easily generated from the confidence score vector. In this work, we will discuss how to evaluate a classification model's efficacy using cross-entropy loss, which is a

$$l(y, f(x;\omega)) = -\sum_{m=1}^{M} 1(y = m) log(p_m(x, \omega)) \tag{4.1}$$

where 1(.) is the indicator function.

4.1.1. Global Model Update. During FL round t, server compiles the uploaded local client models from round t-1. In such case, every client takes part in every training session, $K^t \sqsubseteq$ N represents the sampled clients for local training in round t. The local client model ω_k^t is trained in round t for client $k \in K^t$. Typically, a global model is created by averaging local client models on the server.

$$\omega^{-t} = \frac{1}{|K^{t-1}|} \sum_{k \in K^{t-1}} \omega_k^{t-1} \tag{4.2}$$

In FedDSL, the server trains an average global model on its labeled dataset for the purpose of enhancing it, instead of merely relaying it to the clients for local training in the current round. The technique of data augmentation for image can be utilised to increase the size of the dataset as well as its quality. This is possible because the server only has a limited amount of training data available. The enhanced version of the original input x is denoted by a(x) if we consider the augmentation process a(.). In FedDSL, for instance, we execute weak augmentation using basic operations like random image flipping and random image cropping. Be aware that different learning rounds may provide different augmented images a(x) of the same input x due to the

random nature of the augmentation. The server loss function (using slightly misleading nomenclature) is defined as follows when weak augmentation is applied:

$$L_s(\omega) = \frac{1}{D_i} \sum_{i \in D_l} l(y_i, f(a(x_i); \omega)) \tag{4.3}$$

as previously stated, the cross-entropy loss is denoted as $l(y_i, f(a(x_i); \omega))$. The server continues by running B_s mini-batch gradient descent epochs E_s, which update the global model. Here are the updates made to the global model for each mini-batch step b:

$$\omega^{t,b} = \omega^{t,b-1} - v_s \bigtriangledown_\omega L_s(\omega)|\omega^{t,b-1} \tag{4.4}$$

the server learning rate is denoted as v_s, the initial model is $\omega^{t,0}$=ω^{-t}, and the gradient of $L_s(\omega)$ assessed at $\omega^{t,b}$ is $\bigtriangledown_\omega L_s(\omega)|\omega^{t,b}$.

4.1.2. Calculation of Confidence Threshold. Before clients can use the updated global model to filter input from the unsupervised learning step, $m \in M$ is a confidence threshold for each class determined by the server. Consider the confidence threshold vector $\tau^t = [\tau_1^t ....\tau_m^t]$ as an example, where $\tau_m^t \in [0, 1]$ represents the confidence threshold for class m in round t. Here is the formula for determining the value of τ_m^t.

$$\tau_m^t = \frac{\sum_{i=D_l^{val}} p_m(x, \omega^t) 1(f(x_i; \omega^t) = m)}{\sum_{i \in D_l^{val}} (y_i = m)} \tag{4.5}$$

When it comes to the validation dataset, the denominator represents the total number of instances of class m, while the numerator is the sum of the confidence scores of all the data instances that are classified as class m based on the current model ω^t. For a new data instance x with a high probability of classification result, with the present model ω^t, it is categorized as class m (i.e., $f(x_i; \omega^t) = m$) and the confidence associated with this classification, $p_m(x, \omega^t)$, is more than τ_m^t. Due to the fact that the global model is updated with each new round, τ^t changes across learning rounds. Along with global model at server, clients also download this confidence threshold vector τ^t.

4.1.3. Bootstrapping. Even though clients can begin the process of learning a language with a random beginning model ω^0, our findings indicate that training a robust model on the server labelled dataset and using it as the initial model is more effective in accelerating the convergence of the learning process and achieving higher accuracy.

Due to the utilization of self-ensemble learning in the unsupervised learning phase and the creation of more trustworthy pseudo-labels by means of stronger early rounds models, Bootstrapping has a good influence on convergence. Therefore, FedDSL relies solely on supervised learning at the server with the label dataset to create the initial global model ω^0. Despite the fact that it does not average the local models, it employs equations (3) and (4) as its training basis.

4.2. Unsupervised Learning at the Clients. Clients k who are selected to take part in local client-side training execute unsupervised learning on their unlabeled datasets to create local models ω_k^t receiving the global model ω^t. Unsupervised learning at clients is demonstrated in Figure 4.2

4.2.1. Diverse Self-Ensemble Learning. To generate a pseudo-label, also known as a label prediction, for each instance of unlabeled data using the current model is the objective of effective learning, just as it is in typical SSL problems. Afterwards, construct a dataset in such a way that the pseudo-label of every data instance is accurate on a consistent basis. The term "data filtering" refers to the process of generating a new dataset by selecting records from an existing dataset. Given that the pseudo-label is assumed to be accurate in most cases, it is reasonable to assume that training the model on the filtered dataset would result in improved outcomes.

The dispersed nature of the labeled and unlabeled datasets makes SSFL considerably more difficult in a decentralized context than in a centralized one. Both supervised and unsupervised learning are performed concurrently with one another in conventional SSL. In particular, the model is trained using a single loss

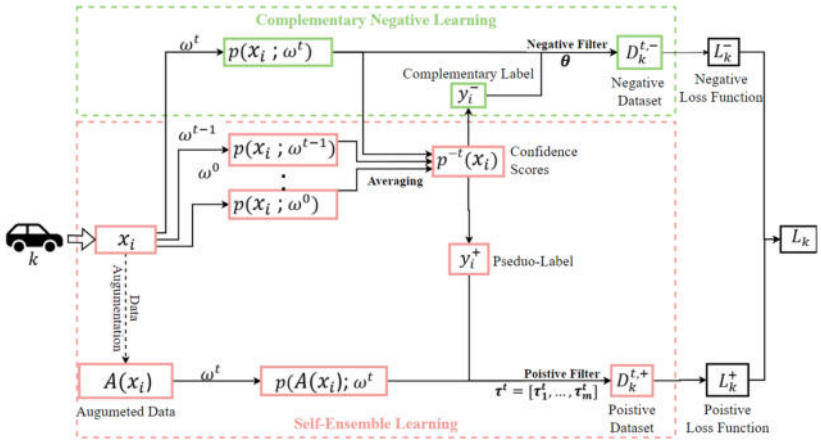

Fig. 4.2: Self-Ensemble Learning Generates the Positive Dataset and Complementary Negative Learning Generates the Negative Dataset and Local Model Update with both datasets.

function Loss(ω), $Loss_L(\omega)$ for which there is a loss on the data that has been labelled and a loss on the data that was not labelled is $Loss_U(\omega)$.

$$Loss(\omega) = Loss_L(\omega) + \alpha\ Loss_U(\omega) \tag{4.6}$$

where the parameter for weight is denoted by α. The filtered dataset can have numerous occurrences with incorrect pseudo-labels due to the low accuracy of the model in the early stages of training. For the purpose of lowering the impact of inaccurate pseudo-labels, a minimal initial value is employed to increase reliance on the trustworthy label dataset during model training. The filtered unlabeled dataset can be given additional weight with an increasing α as the model improves with time.

In SSFL, though, things are different; to train, the client has access to only the loss function applied to the unlabeled data. Therefore, it is not possible to influence whether labeled or unlabeled data is prioritized using this parameter. Consequently, both the local model's performance and the global model's performance after aggregation can be severely impaired because the less-than-ideal global model produced a relatively large number of false pseudo-labels in the first rounds. False positives will build up and spread across iterations of the learning process, rendering the process useless.

An issue that develops when the global model uses inaccurate pseudo-labels is its overconfidence in prediction results. One solution to this problem is self-ensemble learning, which takes advantage of many global models' worth of past data to generate pseudo-labels. This group's judgement mitigates the "stubbornness" of an individual model since it is derived from the same learning procedure, though from different iterations. By using this method, we may decrease the likelihood of overconfidence when assigning classes to data that does not follow a normal distribution and obtain more accurate predictions.

Based on the historical models (ω^0,ω^1,....,ω^t), the client k determines the average score for the confidence vector for each instance $i \in D_k$ in the following way:

$$p^{-t}(x_i) = \frac{1}{t}\sum_{j=1}^{t} p(x_i;\omega^j) \tag{4.7}$$

Client k doesn't have to keep track of every historical model because $p^{-t}(x_i)$ can be updated gradually.

$$p^{-t}(x_i) = \frac{t-1}{t}p^{-t-1}(x_i) + \frac{1}{t}p(x_i;\omega^t) \tag{4.8}$$

Additionally, the pseudo label y_i^+ of x_i is produced based on

$$y_i^+ = argmax_m\ p_m^{-t}(x_i) \tag{4.9}$$

whose corresponding confidence score is $p^{-t}y_i^+(x_i)$. After the server calculates the confidence threshold vector τ^t, the client uses it to filter its local dataset and creates a filtered dataset:

$$D_k^{t,+} = ((x_i, y_i^+); i \in D_k \;\; and \;\; p_{y_i^+}^{-t}(x_i) \geq \; \tau_{y_i^+}^t) \tag{4.10}$$

In order to exclude occurrences without a high level of confidence, the condition $p_{y_i^+}^{-t}(x_i) \geq \; \tau_{y_i^+}^t)$ is used.

4.2.2. Complementary Negative Learning. Data filtering is improved by self-ensemble learning because the use of appropriate pseudo-labels increases the likelihood of including instances in the dataset that has been filtered as $D_k^{t,+}$. Early self-ensemble models had low accuracy due to limited historical models, leading to a significant number of cases with incorrect pseudo-labels being filtered into $D_k^{t,+}$. We apply complementary negative learning to enhance supervised learning for clients, particularly in beginning rounds, as originally promoted for noisy label problems. While classifying instances can be challenging, it is often simpler to eliminate them from incorrect classes. Providing additional information for the purpose of updating the local model can be accomplished by assigning complementary labels to instances, such as classes that are not the true class. Because of this information, the unfavourable impact of incorrect pseudo-labels $D_k^{t,+}$ can be mitigated, resulting in improved performance in local training.

In pursuit of this goal, whilst creating $D_k^{t,+}$, We also generate a second dataset that has been filtered $D_k^{t,-}$, which includes data instances with corresponding labels. In this context, the superscript "+" denotes that the pseudo-labels should represent the actual class, whereas the "-" signifies that the complementary labels should represent any class other than the actual class. Specifically, for every instance $i \in D_k$, according to the results of the averaging eqn. (8) confidence scores for those classes of client k determined, contain a value that is lower than a predetermined threshold that is relatively low. Then, unless it's an empty set, client k randomly selects one from corresponding label for instance i.

$$y_i^- \in (m : p_m^{-t}(x_i) \leq \theta) \tag{4.11}$$

Consequently, the complementary data set is

$$D_k^{t,-} = ((x_i, y_i^-); i \in D_k \;\; and \;\; \ni m, s, t. \;\; (p_m^{-t}(x_i) \leq \theta) \;\; and \;\; p_{y_i^+}^{-t}(x_i) \geq \; \tau_{y_i^+}^t) \tag{4.12}$$

The equation $p_{y_i^+}^{-t}(x_i) \geq \; \tau_{y_i^+}^t)$ guarantees that $D_k^{t,-}$, does not contain an instance that is already in $D_k^{t,+}$.

4.2.3. Local Model Update. To update the local client model, we consider a client loss function with positive and negative components.

$$L_k(\omega) = \lambda \; L_k^+(\omega) + L_k^-(\omega) \tag{4.13}$$

where

$$L_k^+(\omega) = \frac{1}{|D_k^{t,+}|} \sum_{i \in D_k^{t,+}} l(y_i^+, f(A(x_i); \omega)) \tag{4.14}$$

$$L_k^-(\omega) = \frac{1}{|D_k^{t,-}|} \sum_{i \in D_k^{t,-}} l(y_i^-, 1 - f(x_i; \omega)) \tag{4.15}$$

The weight parameter is denoted by the expression $\lambda > 0$. To achieve combine the consistency regularization and pseudo-labeling, making sure to take note of considering that positive loss component $L_k^+(\omega)$ is calculated with the use of strong data augmentation and pseudo-labels A(.) on the input data. In FedDSL, we use a technique known as RandAugment [19], which is a powerful data augmentation method. The weight λ is chosen to be minimal in order to limit the risk that is caused by wrong pseudo labeling. This is because, during the initial rounds, the correct rate for complementary labelling is much higher than that of pseudo labelling.

Table 5.1: Algorithm 1: FedDSL

Server trains the initial model ω^0 based on its labeled dataset
for each FL round t do
Step 1: Server Training
Model averaging to obtain ω^{-t}
Update global model ω^t
Calculate confidence threshold vector τ^t
Step 2: Model Download
All clients download global model ω^t and confidence threshold vector τ^t from the federated central server
Step 3: Client Training
Sample client subset $K^t \sqsubseteq$ N
for each client k **do**
Construct filtered positive dataset $D_k^{t,+}$
Construct filtered negative dataset $D_k^{t,-}$
Update client local model $\omega_k^{t,+}$ according to $D_k^{t,+}$ and $D_k^{t,-}$
end for
Step 4: Upload the Updated Model Upload local client models $\omega_k^t, \forall\ k \in K^t$ to the server
end for

In subsequent rounds, as the historical ensemble creates the pseudo-labels at a higher correct rate, the value λ gradually increases, causing the client loss function to place a greater focus on the positive loss component during those rounds. By carrying out a series of E_c epochs of mini-batch gradient descent with a mini-batch size of B_c, client k is able to update its local model using the client k loss function that was created before. Each mini-batch update is executed by Step b:

$$\omega^{t,b} = \omega_k^{t,b-1} - v_c \bigtriangledown_\omega L_k(\omega)|\omega_k^{t,b} \tag{4.16}$$

$\bigtriangledown_\omega L_k(\omega)|\omega_k^{t,b}$ is the gradient of $L_k(\omega)$ assessed at $\omega_k^{t,b}, v_c$ is the client learning rate, and the initial model $\omega_k^{t,0}$ is the received global model ω^t. One round of SSFL is completed by uploading the obtained local model ω_k^t to the server. Algorithm.1 summarizes the entire FedDSL algorithm.

5. Experiments and Result.

5.1. Experiment Setup. The BDD100K [8] dataset, which contains 100K driving films captured in diverse United States locations and different weather conditions, was used to assess the efficacy of our approach. The movies are 40 seconds long and recorded at 720p and 30 frames per second. They contain GPS and inertial measurement unit data that can be used to plot driving routes. Cloudy, wet, overcast, and snowy weather were the four conditions from which we selected 25K data points for our studies. Our study focuses on five primary categories of objects: humans, cars, buses, trucks, and traffic signs. The dataset is divided up according to different weather conditions in order to model clients with data that is heterogeneous. The testing of our framework in real-world scenarios allows us to study the effects of data heterogeneity on its resilience.

Cityscape [9] can run more experiments with the Cityscapes dataset, which contains street scenes from 50 cities. This dataset lacks precise weather information for each annotation, so clients receive it uniformly randomly. This is our investigation package. It includes 3,525 dummy-annotated test images and 3,475 fine-annotated training and validation images. We also included the second package with 19,998 8-bit images for learning.

Table 5.2 shows our method's performance against baselines and state-of-the-art approaches on BDD100K [8].

5.2. Result. Our trials use one server and multiple clients, depending on the experiment. Each cycle, the server and clients use one local epoch. FedDSL initial conditions include a decay rate of 0.995, local momentum of 0.9, and a complementary threshold θ for negative learning of 0.05. A 0.9 confidence score threshold and 0.001 server learning rate are set. The above parameters are being considered to evaluate test accuracy for

Table 5.2: Our method's performance against baselines and state-of-the-art approaches

Type	Algorithm	Method	Non-IID					IID				
			Cloudy	Overcast	Rainy	Snowy	Total	Cloudy	Overcast	Rainy	Snowy	Total
Centralized	SL	Fully Supervised	0.600	0.604	0.617	0.597	0.605	0.600	0.604	0.617	0.597	0.605
		Partially Supervised	0.540	0.545	0.484	0.474	0.511	0.528	0.545	0.533	0.510	0.529
Federated	SFL	Fully Supervised	0.627	0.614	0.607	0.585	0.608	0.635	0.612	0.608	0.595	0.613
	SSFL	FedAvg [20]	0.560	0.566	0.553	0.553	0.558	0.572	0.588	0.593	**0.610**	0.591
		FedDyn [21]	0.508	0.569	0.541	0.522	0.535	0.355	0.414	0.420	0.397	0.400
		FedOpt [22]	0.561	0.572	0.565	0.566	0.566	0.591	0.587	0.588	0.577	0.586
		FedSTO [7]	0.596	0.607	0.590	0.580	0.593	0.591	0.634	0.614	0.595	0.609
		FedDSL	**0.601**	**0.612**	**0.595**	**0.585**	**0.598**	**0.596**	**0.639**	**0.619**	0.600	**0.614**

Table 5.3: FedDSL's prominence on the Cityscapes [9] datase

Type	Algorithm	Method	Labeled						Unlabeled					
				Categories										
			Person	Car	Bus	Truck	Traffic Sign	Total	Person	Car	Bus	Truck	Traffic Sign	Total
Centralized	SL	Fully Supervised	0.569	0.778	0.530	0.307	0.500	0.537	0.560	0.788	0.571	0.283	0.510	0.542
		Partially Supervised	0.380	0.683	0.193	0.302	0.246	0.361	0.358	0.648	0.343	0.138	0.255	0.348
Federated	SFL	Fully Supervised	0.498	0.715	0.357	0.289	0.410	0.454	0.492	0.714	0.451	0.251	0.425	0.467
	SSFL	FedAvg [20]	0.450	0.697	0.310	**0.304**	0.356	0.423	0.482	0.725	0.425	**0.247**	0.397	0.455
		FedBN [23]	0.488	0.709	0.325	0.285	0.411	0.444	0.375	0.618	0.046	0.031	0.286	0.271
		FedSTO [7]	0.504	0.720	0.342	0.261	0.415	0.448	0.487	0.740	0.460	0.181	0.437	0.461
		FedDSL	**0.510**	**0.724**	**0.348**	0.267	**0.421**	**0.453**	**0.492**	**0.746**	**0.466**	0.185	**0.453**	**0.466**

IID and non-IID scenarios. Table 5.2 shows test accuracy for IID and non-IID scenarios for various methods. For labelling client instances, Fully Supervised Learning has the highest accuracy and performance limit. the efficiency with which FedDSL uses client unlabeled data. FedDSL outperforms competitors in all weather and data distribution scenarios (IID and Non-IID). A lot of unlabeled data may have incorrect pseudo labels. Raising the confidence score threshold reduces unlabeled data but improves pseudo-labeling accuracy.

Our FedDSL method always outperforms other Semi-Supervised FL methods. Its better results in difficult Non-IID situations demonstrate its resilience to data distribution changes. Our method performs well under IID conditions, producing results comparable to a fully supervised centralised approach. These studies demonstrate that FedDSL maximises FL's benefits while minimising its drawbacks. Our approach's good performance in different weather and data distributions suggests real-world use.

Table 5.3 showcases FedDSL's prominence on the Cityscapes [9] dataset across different object types.

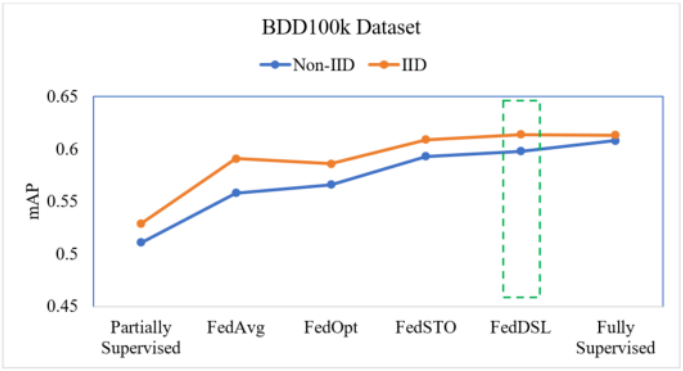

Fig. 5.1: Evaluation of different approaches on BDD100k [8] Dataset

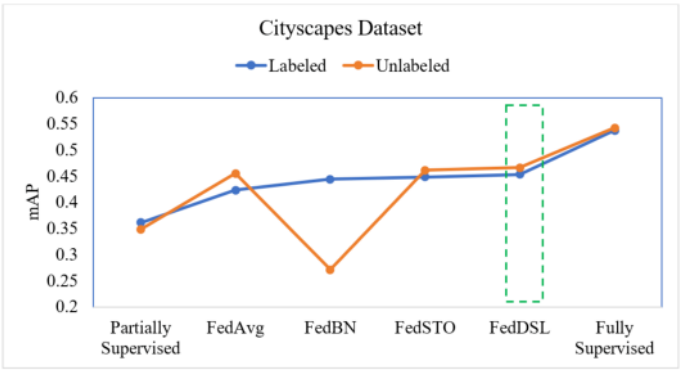

Fig. 5.2: Efficiency in diverse aspects using Cityscapes [9] dataset labelled and unlabeled data

5.3. Evaluation with mAP@0.50. FedDSL's effectiveness is demonstrated by the mAP@0.50 algorithm when applied to the BDD100K[8] dataset (Table 5.2). The Fully Supervised Centralized approach has an average measure of performance (mAP) of 0.605 in Non-IID scenarios, whereas the FedDSL approach has a mAP of 0.598. Under IID conditions, the mAP of FedDSL is 0.613, which is comparable to the mAP of SSFL, which is 0.614. Based on the findings, it appears that FedDSL is capable of detecting objects just like other systems. A comparison of our method and others' performance on the BDD100K[8] Dataset is presented in Figure 5.1.

Table 5.3 displays the results of applying the mAP@0.50 method to the Cityscapes dataset, this demonstrates that the FedDSL technique is effective. Results from the two approaches is similar, in Labeled situations, the Fully Supervised Centralized method achieves an average mAP of 0.537 while FedDSL recorded 0.453. Meanwhile, under Unlabeled conditions, FedDSL finds a mAP of 0.466, which is quite close to the value of 0.467 found by FSL(Fully Supervised Learning). It appears from the results that FedDSL can detect objects just as well as other systems. Figure 5.2 represents the performance across different object types on the Cityscapes[9] Dataset.

6. Conclusion. As a machine learning framework, FL shows a lot of potential in addressing data privacy and decentralised datasets. The novel Semi-Supervised Federated Learning framework, with its distinctive ensemble learning technique called FedDSL, is introduced in this research. Using self-ensemble learning and negative learning with personalized pseudo labelling, FedDSL is built to tackle the difficulties of federated learning with diverse unlabeled data. It may coordinate server-side supervised learning with client-side unsupervised learning. These methods improve object detection capabilities in many different settings and data distributions by enabling powerful, diverse feature learning. FedDSL has been found to beat previous Semi-

Supervised Federated Learning techniques by a significant margin, thus indicating that it is possible to use unlabeled data to enhance learning even in the Federated Learning scenario. Extensive experimental validation confirmed that BDD100K dataset has an average mAP of mAP of FedDSL is 0.613 and in Cityscapes dataset under Unlabeled conditions, FedDSL finds a mAP of 0.466 which is quite close to FSL. Significant progress toward more efficient and privacy-conscious learning in real-world Federated Learning contexts has been made with this accomplishment. It is our sincere wish that this paper's findings will encourage more investigation into this promising area.

REFERENCES

[1] Truong, N., Sun, K., Wang, S., Guitton, F. and Guo, Y., 2021, Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Computers & Security*, *110*, p.102402.

[2] Blanco-Justicia, A., Domingo-Ferrer, J., Martínez, S., Sánchez, D., Flanagan, A. and Tan, K.E., 2021, Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence*, *106*, p.104468.

[3] Jin, Y., Wei, X., Liu, Y. and Yang, Q., 2020, Towards utilizing unlabeled data in federated learning: A survey and prospective. *arXiv preprint arXiv:2002.11545*.

[4] Jin, Y., Liu, Y., Chen, K. and Yang, Q., 2023, Federated Learning without Full Labels: A Survey. *arXiv preprint arXiv:2303.14453*.

[5] Diao, E., Ding, J. and Tarokh, V., 2022, SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training. *Advances in Neural Information Processing Systems*, *35*, pp.17871-17884.

[6] Bian, J., Fu, Z. and Fedseal, J.X., 2021. Semi-supervised federated learning with self-ensemble learning and negative learning. *arXiv preprint arXiv:2110.07829*.

[7] Kim, T., Lin, E., Lee, J., Lau, C. and Mugunthan, V., 2023, Navigating Data Heterogeneity in Federated Learning: A Semi-Supervised Approach for Object Detection. *arXiv preprint arXiv:2310.17097*.

[8] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V. and Darrell, T., 2020, Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2636-2645).

[9] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B., 2016, The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213-3223).

[10] Sakaridis, C., Dai, D. and Van Gool, L., 2018, Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, *126*, pp.973-992.

[11] Han, J., Liang, X., Xu, H., Chen, K., Hong, L., Mao, J., Ye, C., Zhang, W., Li, Z., Liang, X. and Xu, C., 2021, SODA10M: a large-scale 2D self/semi-supervised object detection dataset for autonomous driving. *arXiv preprint arXiv:2106.11118*.

[12] Du, Z., Wu, C., Yoshinaga, T., Yau, K.L.A., Ji, Y. and Li, J., 2020, Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open Journal of the Computer Society*, *1*, pp.45-61.

[13] Nguyen, D.C., Pham, Q.V., Pathirana, P.N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O. and Hwang, W.J., 2022, Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, *55*, pp.1-37.

[14] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S. and McMahan, H.B., 2020, Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

[15] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A. and Smith, V., 2020, Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, *2*, pp.429-450.

[16] Xu, B., Chen, M., Guan, W. and Hu, L., 2023, Efficient Teacher: Semi-Supervised Object Detection for YOLOv5. *arXiv preprint arXiv:2302.07577*.

[17] Zhou, H., Jiang, F. and Lu, H., 2023, SSDA-YOLO: Semi-supervised domain adaptive YOLO for cross-domain object detection. *Computer Vision and Image Understanding*, *229*, p.103649.

[18] Jeong, W., Yoon, J., Yang, E. and Hwang, S.J., 2020, Federated semi-supervised learning with inter-client consistency & disjoint learning. *arXiv preprint arXiv:2006.12097*.

[19] Cubuk, E.D., Zoph, B., Shlens, J. and Le, Q.R., Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 702-703).

[20] McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B.A., 2017, Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282) PMLR.

[21] Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N. and Saligrama, V., 2021, Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*.

[22] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S. and McMahan, H.B., 2020, Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

[23] Li, X., Jiang, M., Zhang, X., Kamp, M. and Dou, Q., 2021, Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*.

[24] Zheng, T., Li, A., Chen, Z., Wang, H. and Luo, J., 2023, AutoFed: Heterogeneity-Aware Federated Multimodal Learning for Robust Autonomous Driving. arXiv preprint arXiv:2302.08646.

[25] Pokharel, P. and Dawadi, B.R., 2023, Federated Machine Learning for Self-driving Car and Minimizing Data Heterogeneity

Effect. In International Conference on Computing and Information Technology (pp. 41-52) Cham: Springer Nature Switzerland

Edited by: Mahesh T R
Special issue on: Scalable Dew Computing for future generation IoT systems
Received: Feb 1, 2024
Accepted: May 1, 2024