The OARF Benchmark Suite: Characterization and Implications for Federated Learning Systems

Sixu Hu, Yuan Li, Xu Liu, Qinbin Li, Zhaomin Wu, Bingsheng He
School of Computing
National University of Singapore
13 Computing Drive Singapore 117417
{sixuhu, liyuan, liuxu, qinbin, zhaomin, hebs}@comp.nus.edu.sg

Abstract

This paper presents and characterizes an Open Application Repository for Federated Learning (OARF), a benchmark suite for federated machine learning systems. Previously available benchmarks for federated learning have focused mainly on synthetic datasets and use a very limited number of applications. OARF includes different data partitioning methods (horizontal, vertical and hybrid) as well as emerging applications in image, text and structured data, which represent different scenarios in federated learning. Our characterization shows that the benchmark suite is diverse in data size, distribution, feature distribution and learning task complexity. We have developed reference implementations, and evaluated the important aspects of federated learning, including model accuracy, communication cost, differential privacy, secure multiparty computation and vertical federated learning.

1 Introduction

Federated Learning (FL), first introduced by McMahan et al. [43], is a technique that enables multiple parties to train a model collaboratively without leaking their private data. Recently, FL has become a hot research topic in both industry and academia [61, 35]. Various machine learning models, communication methods, privacy-preserving methods and data splitting schemes have been researched under federated settings. Despite of the success in those research and development, a new benchmark suite is urgently needed to study and compare various FL designs, and help guide the design and implementation of future FL systems.

Looking back in history, we observe that benchmarks have played an important role in the machine learning area. Benchmark suites like MLPerf [49, 41] and DAWNBench [13] have provided various benchmark metrics and results for deep learning training and inference. These benchmarks facilitate the comparison between machine learning frameworks and models. However, compared to the machine learning area, we found that there lacks a federated learning benchmark for researchers and developers to reference.

There already exist some preliminary benchmarks [9, 46, 39, 37] targeting FL applications. However, these benchmarks only include non-federated datasets, where datasets are artificially split from a single dataset, which can be unrealistic scenarios. The types of models and metrics they have considered are also limited. For example, benchmark by Nilsson et al. [46] provides a performance evaluation method based on Bayesian correlated t-test, but it only focuses on how communication architecture and data distribution affect the model accuracy.

A good benchmark for FL systems needs to address a series of factors. First, it needs to reflect the real-world scenarios by providing dedicated datasets and workloads. Second, it needs to include

Preprint. Under review.

various metrics and measuring methods to depict a full picture of FL systems. Recent surveys on federated learning [33, 35] suggested that in addition to the issues already exist in the machine learning area, there are at least data partitioning, privacy, communication, fairness issues that need to be identified.

Taking these factors into consideration, we present OARF, a benchmark suite that aims to evaluate and study the properties of different FL systems, and provides tools to help design next generation FL platforms. Our work highlights the following contributions.

We assemble representative datasets and design representative workloads to reflect real-world scenario. In real-world scenarios, the datasets are often from different parties and heterogeneous. We have collected and assembled real-world datasets from different sources and designed workloads covering numerous domains to reflect this situation.

We study the intrinsic properties of federated learning and its components. We study the intrinsic properties of FL systems, including the relations between various design metrics such as data partitioning, privacy mechanisms and machine learning models. These properties enrich our knowledge of the internal mechanism for federated learning, provide valuable experience for the industry needs, and provide suggestions on building future FL frameworks.

We provide reference implementations and observe interesting findings. We provide reference implementations to our studies for better reproducibility and modularity. Each reference implementation evaluates one or more properties stated above. We conclude the common evaluation results of these implementations as our findings to the intrinsic properties of FL system and its components. One finding is that, with careful design, FL systems achieve very small accuracy loss compared with centralized training (without privacy constraints). The other is, homomorphic encryption, which is widely used in numerous FL algorithm design, is indeed very costly in computation and it should be used only when necessary.

2 Background and Related Work

2.1 Federated Learning

Recent surveys by Li et al. [35] and Yang et al. [61] indicate that federated learning is a combination of various techniques in multiple areas, including data partitioning, machine learning model, privacy and communication. Designing a good benchmark that measures the performance of all these aspects and their mutual effects is inherently challenging. Specificially, we consider the following properties: 1) *Models:* Existing efforts have developed federated algorithms for neural networks [55, 43, 54], tree-based models [34, 11], linear models [26, 45] or other types of models. 2) *Data partitioning* is another important concept that needs to be considered in our work. [61] introduced three partition schemes in their work: horizontal, vertical and hybrid, where horizontal FL uses datasets that share feature space but not sample space, vertical FL uses datasets that share sample space but different in feature space, and hybrid FL is the combination of both. 3) *Security and Privacy:* Secure multiparty computation (SMC) [62], homomorphic encryption [22] and differential privacy [19] are widely used to protect the security in the training, communication and model release process. 4) *Communication architectures* [35] and communication cost [55] are two of the major concerns in FL systems due to their impacts on model performance.

There have been already some newly emerged FL platforms, and Li et al. [35] summarized most of them, namely Federated AI Technology Enabler (FATE) [2], TensorFlow Federated (TFF) [5], PySyft [3] and PaddleFL [4]. However, these platforms are still quite immature and lack basic functionalities. Neural networks and linear models are supported by all of the four frameworks, while PySyft, TFF and PaddleFL do not support decision trees currently. For data partitioning, only FATE has implemented models and utilities for both vertical and horizontal partitioning currently, while the others only support the horizontal setting. As for privacy mechanisms, PySyft and PaddleFL provide more choices compared to the other two. Finally, none of the current frameworks implements tools or protocols for decentralized communication architecture.

2.2 Federated Learning Benchmarks

There have been some initial efforts for benchmarking FL systems. Table 1 summarizes the characteristics of these benchmark-related works. To the best of our knowledge, there is no benchmark that is using federated datasets currently, where real datasets in different parties come from different sources. There is no benchmark that is comprehensive enough to evaluate the four important aspects of federated learning.

Table 1: Comparison of FL Benchmarks

Name	Federated Dataset	Partitioning Scheme	Various ML Models	Privacy Mechanism	Comm. Cost
LEAF [9]	✗	✗	✓	✓	✓
Nilsson et al. [46]	✗	✗	✗	✗	✓
Street Image [39]	✗	✗	✓	✗	✓
Edge AIBench [25]	✗	✗	✓	✗	✗
Liu et al. [37]	✗	✗	✗	✗	✓
OARF (our work)	✓	✓	✓	✓	✓

We introduce more details about existing FL benchmarks. LEAF [9] provides several image/text datasets and a set of reference implementations using federated averaging. However, LEAF only focuses on the massively cross-device scenario, where federated learning is performed on a massive number of devices, but neglects the cross-silo scenarios. Nilsson et al. [46] have proposed a benchmark using Bayesian t-test to measure the performance of different FL algorithms such as Federated Averaging (FedAvg). Luo et al. [39] have proposed a street image dataset for federated learning to provide high-quality labeled data for FL research. They evaluated accuracy and communication costs of YOLO and Faster R-CNN under different settings. Liu et al. [37] have introduced an evaluation framework for large-scale benchmark which focuses on communication costs.

3 The OARF Benchmark Suite

3.1 OARF Design Principles

This paper aims at providing researchers and developers with a comprehensive, easy-to-use benchmark suite, with the following design principles. First, **Emerging Frameworks.** To keep up with the rapid development of FL, we carefully select datasets, workloads and reference implementations in order to have a high relevance to the state-of-the-art in FL. Second, **Diversity of Workloads.** Federated learning can be used in a number of domains, such as biology, finance and mobile applications. Tasks in different domains vary greatly from each other. Data formats the machine learning models vary from task to task. For different tasks, the data distribution and data partitioning scheme are also different. Although it is impractical to design a benchmark to cover all kinds of tasks, efforts are made in OARF to maximize the diversity of workload, so that different types of tasks and the corresponding applications can be characterized. Third, **Comprehensiveness.** Federated learning can be characterized in various ways. As state in Section 2.1, it can be at least categorized by: 1) Partitioning scheme, including horizontal, vertical and hybrid federated learning. 2) Underlying models, including decision trees, different types of neural networks and other machine learning model as the underlying model. 3) Privacy and security mechanisms, including secure multiparty computation and differential privacy. 4) Communication architectures, including synchronous/asynchronous, as well as centralized/decentralized design. To cover all these aspects, OARF uses various metrics to measure the performance of each component in federated learning, and explore how they contribute to the entire learning process. Fourth, **Openness.** Since federated learning is a rapidly developing field, we make OARF open to modification and addition. OARF users can easily contribute or modify different parts of OARF to meet their specific requirements.

3.2 Overview

Figure 1 shows the structure of the OARF benchmark. It adopts a layered design, with three layers namely Metrics, Tasks (Workloads), and Reference Implementations. Each layer can be extended or enhanced accordingly. We pick tasks according to the properties of these aspects, with respect to the partitioning scheme, the domain and models. A task and one or more metrics are combined into a reference implementation to reveal the intrinsic properties.

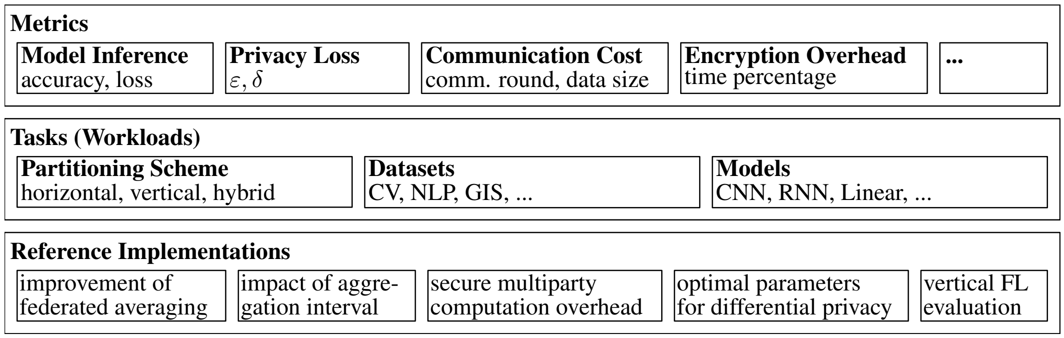


Figure 1: OARF benchmark components

The components in the benchmark suite are carefully chosen to achieve our design principles: 1) Emerging workloads: we use up-to-date datasets like Video Game Sales and up-to-date federated learning framework like PySyft for our training. 2) Diversity: we use text, image and structured data in our datasets, and cover different domains, including CV and NLP. Our tasks use various models, including CNN and RNN. 3) Comprehensiveness: Our tasks consist of both horizontal and vertical federated learning, and we benchmark each task from multiple aspects. 4) Openness: All our reference implementations are publicly available and open to modification. The following parts of this section elaborate on the tasks we provide in the OARF benchmark suite.

3.3 Metrics

Based on the major components of FL systems, we provide four types of metrics to measure a FL system. For the model inference part, we use accuracy and loss that are often considered in machine learning benchmarks [49, 13]. For privacy measurement, we adopt the definition of (ε, δ)-differential privacy [6] to describe the privacy loss. The communication cost is captured by both communication rounds and amount of data transferred in each communication. Finally, encryption overhead is described by the extra in training time percentage.

3.4 Workloads and Reference Implementations

To cover various scenarios of federated learning, we collected public available datasets, and categorize them by data partition schemes and tasks, as shown in Table 2. Different datasets belonging to the same task can be used to simulate data possessed by different parties. All these datasets are from different sources, which means that they are "naturally" owned by different parties instead of split from a single dataset. Datasets like these more precisely reflect how data is distributed in real-world federated learning tasks.

In the following, we only describe the tasks and datasets that are demonstrated in the experiment section. A complete list of descriptions are shown in the supplementary materials.

Chinese Character Recognition Handwritten character recognition has been extensively studied for decades and achieved impressive progress with the emergence of deep learning. Instead of alphanumeric characters, in this task we study and predict Chinese characters. It is more challenging than alphanumeric character recognition due to a larger number of categories and more complex character structures. In our setting, new challenges arise due to various image quality and handwriting styles on different datasets. Works by Zhong et al. [66] and Xiao et al. [60] present CNN structures and variations that reach state-of-the-art results on the problem.

Reference implementation: We set up our federated task with datasets CASIA-HWDB1.1 and HIT-OR3C and use a variation of VGG structure [52] as our underlying algorithm. We extend the training part to the federated version using federated averaging, where we train two identical models and aggregate their weights every epoch. The task is trained on a CNN model that consists of 11 convolutional units, a 3-layer MLP and a softmax layer for the prediction. For both of the datasets,

Table 2: A list of tasks and corresponding datasets. In the title, P[1] is Data Partitioning, and D[2] is Domains. The Steam Game dataset† only counts the number of games, and there is also players' information in this dataset. The IMDB Movie dataset‡ counts only the number of unique titles.

<table>
<tr><th>P[1]</th><th>D[2]</th><th>Task</th><th>Datasets</th><th># instances</th></tr>
<tr><td rowspan="13">Horizontal</td><td rowspan="10">CV</td><td rowspan="5">Gender / Age Prediction</td><td>All-Age-Face [10]</td><td>~13k</td></tr>
<tr><td>APPA-REAL [7, 12]</td><td>~7.6k</td></tr>
<tr><td>OUI Audience Face [21]</td><td>~26k</td></tr>
<tr><td>IMDB-WIKI – 500k [51]</td><td>~52k</td></tr>
<tr><td>Labeled Faces in the Wild [31, 30]</td><td>~13k</td></tr>
<tr><td rowspan="2">Face Recognition</td><td>BUPT-Balancedface [56, 57, 58]</td><td>~1.25M</td></tr>
<tr><td>Racial Faces in-the-Wild [56, 57, 58]</td><td>~40k</td></tr>
<tr><td>Alphanumeric Character Recognition</td><td>Chars74K [15]</td><td>~74k</td></tr>
<tr><td rowspan="2">Chinese Character Recognition</td><td>HIT-OR3C [67]</td><td>~460k</td></tr>
<tr><td>CASIA-HWDB1.1 [53]</td><td>~1.1M</td></tr>
<tr><td rowspan="2">NLP</td><td rowspan="2">Sentiment Analysis</td><td>IMDB Movie Review [40]</td><td>~50k</td></tr>
<tr><td>Amazon Movie Review [42]</td><td>~8M</td></tr>
<tr><td rowspan="1">GIS</td><td>Traffic Prediction</td><td>METR-LA [32]
PEMS-BAY [36]</td><td>~34k
~52k</td></tr>
<tr><td rowspan="8">Vertical / Hybrid</td><td rowspan="8">General ML</td><td rowspan="3">Trend Prediction / Recommendation</td><td>Steam Game [47]</td><td>~17k†</td></tr>
<tr><td>IGN Rating [1]</td><td>~18k</td></tr>
<tr><td>Video Game Sales [1]</td><td>~55k</td></tr>
<tr><td rowspan="2">Trend Prediction / Recommendation</td><td>MillionSong [8]</td><td>~1M</td></tr>
<tr><td>Free Music Archive [16]</td><td>~106k</td></tr>
<tr><td rowspan="3">Trend Prediction / Recommendation</td><td>MovieLens 1M [27]</td><td>~1M</td></tr>
<tr><td>Movie Industry [23]</td><td>~6.8k</td></tr>
<tr><td>IMDB Movie [40]</td><td>~4M‡</td></tr>
</table>

we use 80% of the character writers for training, in which 20% is used for validation. The rest 20% of the data is for testing. As the amounts of data trained on two parties are different, we also assign weights to each parties' model parameters, and the weights are proportional to the number of records in each parties' training dataset respectively.

Sentiment Analysis Sentiment analysis is a technique that uses natural language processing and text analysis to study and predict affective states. The emergence of machine learning has greatly propelled the study in this area. Recent works [59] and surveys [64] have proved that LSTM is effective for this kind of task. Today, various rating websites like IMDB and Rotten Tomato provide a large amount of data for the training of sentiment analysis model.

Reference implementation: We apply federated averaging to an LSTM model, and use movie review data from Amazon and IMDB for the training. Limited by the GPU capacity and training time, We use the whole 50,000 entries in IMDB Movie Review dataset and randomly sampled the same amount of entries from the Amazon Movie Review dataset as the training data. The labels are i.i.d. distributed, where 50% of the sentences are marked as positive and the other 50% negative. For both datasets, we use 80% data for training, 10% for validation and 10% for testing, all of which are identically distributed. The task is trained on LSTM model that consists of an embedding layer with the dimension of 512, two LSTM layers with the hidden dimension of 256, a dropout layer with dropout probability of 0.3, a fully connected layer and a sigmoid layer for the output. The output of the model is a binary label indicating whether the given text segment is positive or negative.

Recommendation Recommendation system has become a core component in various industry applications such as product promotion and advertisement display [48]. For instance, movie recommendation can help a company to give better estimates of its users' preferences. As the three groups of movie datasets listed in Table 2 contain user information and movie information, we can perform federated recommendation based on these datasets.

Reference implementation: We split the MovieLens 1M dataset into two parties, having the rating matrix in one and the auxiliary information of users and movies in the other. The two parties can then be vertically federated to predict a user's preference to a movie. We use a variation of Neural Collaborative Filtering [29] as our underlying model. We implement its federated version using FATE.

4 Experiment

In this section, we present the evaluations to our reference implementations. The evaluations are divided into two parts: 1) For horizontally partitioned tasks, we demonstrate the performance improvement, communication cost, influence of differential privacy and secure multiparty computation in terms of accuracy and efficiency in Section 4.1 through Section 4.4. 2) For the vertically partitioned task in Section 4.5, we mainly focus on the impact of homomorphic encryption method.

The experiments are conducted on a Linux machine with a Xeon E5–2640 CPU @ 2.4GHz, 256GB DRAM and an Nvidia Tesla P100 GPU. The experiments can be reproduced with the code in the supplementary materials.

4.1 Improvement of Federation

Table 3 shows the results for three types of training setups: training each dataset separately, train the combined dataset from all parties, and use federated averaging algorithm [43] to train on the data collaboratively. Accuracy in all these setups are measured with the two types of test datasets: each party's own test dataset and the combined one created by concatenating two parties' test datasets.

The results using combined dataset for testing show a fair comparison between different training setups. In this case, our federated reference implementation outperforms training with each party's private data in both of our tasks, and is close to the training with combined dataset. The results show that federated learning can effectively improve model accuracy, and the accuracy loss can be small in comparison with the centralized training.

When testing each setup with a dataset from two parties respectively, we make two observations: 1) When the training and testing dataset come from the same party, the result describes the model quality. 2) When they come from different parties, the result describes the generalization ability of the model. Federated averaging improves both the quality of the models and the generalization ability of the models, and achieves lower loss than the combined dataset setup.

Table 3: Accuracy and loss at the end of the training. "Val. Loss" for validation loss.

(a) Chinese character recognition

Training Dataset	Test Dataset			Val. Loss
	Combined	CASIA	HIT	
CASIA	93.7%	92.8%	95.7%	0.41
HIT	77.5%	68.5%	97.1%	1.95
Combined	94.8%	93.2%	98.2%	0.33
FedAvg	95.4%	94.1%	98.5%	0.27

(b) Sentiment analysis

Training Dataset	Test Dataset			Val. Loss
	Combined	IMDB	Amazon	
IMDB	82.3%	84.9%	79.5%	0.39
Amazon	83.3%	80.4%	86.1%	0.40
Combined	87.1%	86.6%	87.2%	0.37
FedAvg	86.0%	85.1%	86.7%	0.34

4.2 Communication Cost

We investigate how the model accuracy changes with federation frequency in Figure 2. The accuracy is measured with combined test datasets. Limited by the GPU resource and training time, we fix the number of epochs to be 10 and 30 for the Chinese character recognition and the sentiment analysis, respectively. We define *average internal* as the number of epochs trained between two averages. Overall, the accuracy decreases with the average interval for both task. Compared to the sentiment analysis task, the decrease in the Chinese character recognition task is much smaller, indicating that different models have different sensitivities to the average interval. Using the model size, we can calculate the amount of data transmitted. The size of the weights is 55.3MB and 551.2MB for the Chinese character recognition task and for the sentiment analysis task, respectively. With average interval grows from 1 epoch to 2 epochs, the data transmitted for the two tasks decreases 553MB and 16,536MB respectively, and the accuracy drops from 93.3% to 93.1% and 84.9% to 80.0% respectively. Clearly, there is a trade-off between communication cost and accuracy.

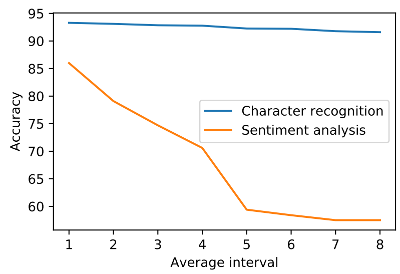


Figure 2: Model accuracy under different communication cost

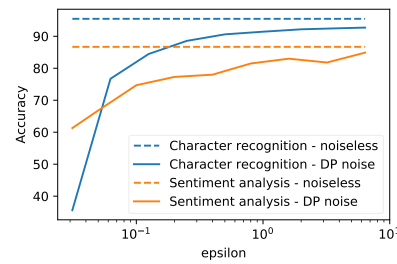


Figure 3: Model accuracy under different privacy cost

4.3 Differential Privacy

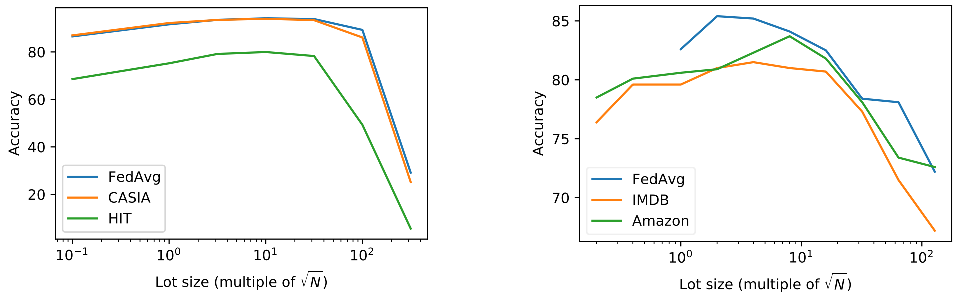
To investigate how to set privacy parameters, we analyze the relationship between different privacy parameters and model quality. There are multiple types of composition mechanism for differential privacy [19, 18, 6, 44], and we use the moments accountant [6] mechanism as it is designed for SGD and imposes better privacy bound than other DP composition mechanisms like simple composition and adaptive composition [20]. TensorFlow provides an `tensorflow_privacy` module that allows us to count how much privacy budget is spent using this mechanism, and we use that module to calculate our noise parameters. As the calculation process requires a fixed epoch number, we fix our epoch number to 20 for both the Chinese character recognition task and the sentiment analysis task, where the models reach a decent accuracy and not too much extra privacy budget needs to be wasted on minor accuracy improvements.

We configure the parameters so the model satisfies (ε, δ)-differential privacy where we fix δ to be the inverse of the number of records in the dataset [18] and vary ε. As shown in Figure 3, the accuracy grows with the privacy budget ε, which matches the intuition that the less noise we add, the model becomes less private but its accuracy increases. When ε grows above 1, the growth of accuracy becomes negligible. The FedAvg curve of the sentiment analysis task is not complete because when lot size is less than $\sqrt{N}$, the federated model does not converge.

Another parameter that has a great impact when adding DP noise, indicated by Abadi et al. [6], is *lot size*, which indicates how much records should be trained before adding a noise. In their work, the empirical optimal lot size is $\sqrt{N}$, which is what we set in Figure 3. To further investigate the impact of lot size on federated learning, we fix ε to 2.0 and change lot size, and the results are shown in Figure 4. From these two figures, all setups of the two tasks reach their peak performance at between $\sqrt{N}$ and $10\sqrt{N}$, which is close to the previous empirical result [6].

4.4 Secure Multiparty Computation

We apply the SPDZ [14] secure multiparty computation technique to encrypt the parameter exchange process, and use PySyft to implement the encryption process. Table 4 shows additional encryption overhead by SMC. The time needed for encryption is proportional to the size of the weight. We also notice that SMC causes a small loss of accuracy. This is due to the SPDZ can only be done on integer values, but our models use floating point weights. Converting between floating-point values and integer values brought precision loss, which further leads to accuracy loss. This loss can be mitigated by using larger integers to store the converted gradient, at a cost of larger communication cost.



(a) Chinese character recognition

(b) Sentiment analysis

Figure 4: Test accuracy under different lot size

Table 4: Test accuracy and training time

(a) Chinese character recognition

Training Technique	Test Dataset		Time Cost per Epoch
	CASIA	HIT	
w/o SMC	95.0%	98.5%	955s
with SMC	94.8%	98.2%	960s

(b) Sentiment Analysis

Training Technique	Test Dataset		Time Cost per Epoch
	IMDB	Amazon	
w/o SMC	85.6%	85.8%	84s
with SMC	84.4%	84.8%	145s

4.5 Homomorphic Encryption in Vertical Federated Learning

We give a detailed analysis of the vertical federated learning task, namely movie recommendation in this section. The mean-squared error result of the experiment is shown in Table 5. The Rating + Auxiliary setting directly combines rating matrix and auxiliary information such as user gender and movie genre, while the Rating fed Auxiliary setting uses homomorphic encryption module in FATE to perform the vertical federated learning. As shown in the result, combining the rating matrix with auxiliary information improves the model performance on the test dataset.

We apply the Heterogeneous Neural Network module in FATE for secure and lossless training because the Neural Collaborative Filtering model can be implemented in it with minor modification. In our experiment, the average spent time of each batch is around 206 seconds and the estimated training time is about 403 hours, which is impractical to train on our local machine. Compared with the non-federated setting, the huge increase of training time is mainly incurred by the homomorphic encryption and homomorphic computation operation, while the communication time is negligible.

Table 5: Test MSE and training time of the movie recommendation task

Train Dataset	Test Dataset MSE	Time per Epoch
Rating	0.7549	14s
Rating + Auxiliary	0.7195	39s
Rating fed Auxiliary with HE	N/A	~403h

5 Conclusions

In this paper, we propose the OARF federated learning benchmark suite, which contains three modules: metrics, tasks and reference implementations. We use this benchmark suite to reveal the internal properties of federated learning and its modules. We have maintained OARF in GitHub (the URL is omitted for anonymity). The reference implementations in the benchmark suite are by no means complete. Besides making this benchmark suite open to the community, we also plan to add more modules and reference implementations in the future.

Through experiments, we show the intrinsic properties of FL systems and their major components. First, we show that federated learning can effectively boost the performance of a model, and achieve very small accuracy loss compared with centralized training (without privacy constraints). Second, we show a method to tune the differential privacy parameters to achieve decent privacy while maintaining reasonable model accuracy. Third, even though widely used in federated learning, the overhead brought by homomorphic encryption is too large to be practical.

Broader Impact

Who may benefit from our research? We propose a new benchmark suite, which sets a standard for analyzing the internals of federated learning. Developers and researchers in the FL field can use our benchmark suite to test their own FL systems and applications, and use the reference implementations as baseline to test the effectiveness of their design. Companies may use our benchmarks to make design decisions in their own model implementations. We believe that a good benchmark will boost the research and development of the entire community in federated learning.

Weak points and future works. Our benchmark has several weak points. One is that the tasks and the corresponding datasets for vertical FL are limited, since most vertical tasks require an entity id for alignment to join two or more datasets, and most public datasets are anonymized so it is hard to find such federated datasets from two different parties. The other is that the metrics we evaluate are quite limited. For example, data skew is concerned by various FL works and can affect the model accuracy to an extent. In the future, we plan to make up these deficiencies, by adding new reference implementations or appealing the community to contribute their datasets and implementations to this project.

Appendix

Description of Tasks and Datasets

Gender/Age Prediction Age and gender prediction have attracted growing interests due to its potential in various areas, including security control, crime prevention and human interaction. These two tasks have been extensively studied by a number of works [24, 21, 50]. The goal of this task is to predict the gender or age from given human face images.

We collected 5 datasets for this task, each containing a number of face images and corresponding gender/age label. These images are fed into a CNN model to train a classification model. Among all the five datasets, the OUI Audience Faces and IMDB-WIKI contain not only portrait photos but also full-body images, so they need a face position detection preprocessing phase to extract the face image. For networks that are designed to take fixed-size images as input, images from different datasets need to be preprocessed to the same size.

In our reference implementation, we used All-Age-Face and APPA-REAL dataset to learn a ResNet-based CNN model collaboratively. We choose these two datasets because both of them contains gender and age label, and both of them provide preprocessed and aligned portrait version.

Face Recognition Face recognition has been a long-standing research topic in the computer vision community and has been widely used in different real-world applications. Extensive research works have been done [65, 17] to continuously improve the state-of-the-art. Face recognition uses human face images as input and learns to identify the identities. Datasets for this task contain face images and the identities corresponding to the faces. For each individual, there are multiple images associated with it. These images are divided into two parts, one for training and another for validation. The testing identities are disjoint from the training and validation sets.

The racial bias phenomenon in face recognition is a problematic issue and has attracted increasing attention. Wang et al. [58] shows that both commercial APIs and SOTA algorithms work unequally well for different races. Wang et al. [58] also proposes a method to solve this problem. In our benchmark, we try to give a solution that makes use of federated learning. Our goal is to train a model that has better generalization ability to identify faces in different races, thus reducing racial bias.

We focus on the task of face verification. We employ the ResNet-34 [28] architecture with softmax loss and train the model with federated averaging on two subsets of BUPT-Balancedface dataset, namely Caucasian-7k and African-7k. These face images are horizontally flipped for data augmentation. For data prepossessing, we generate the similarity transformed faces by utilizing the face landmarks detected by MTCNN [63]. We then crop the faces to 112×112 and normalize them.

In the testing, we use the racial-balanced testing set Racial Faces in-the-Wild to evaluate the capability of the model. We follow [38] to get the feature representation by concatenating an image's original features and its horizontally flipped features, both extracted from the output of the last FC layer. The similarity score is computed by the cosine distance of features from an image pair. Then thresholding and k-fold cross-validation techniques are used for getting the final verification accuracy.

Alphanumeric Character Recognition Handwriting character recognition is a classic and relatively well-studied field. Proposed in 1998, the MNIST dataset remains the most widely known and used dataset. People are easy to achieve 99% accuracy on it by applying deep learning technique. Also, comprehensive experimental results on MNIST in the federated scenario have been given in the Benchmark like LEAF [9], and other federated algorithms [43, 54].

The Chars74k handwritten dataset, however, is still challenging to get a high testing accuracy and has not applied in federated learning. We incorporate two subsets of Chars74k dataset (Fnt and Hnd) in our benchmark. They have a significant difference in terms of dataset size, data quality, and writing shape.

We define the task as to classify images from handwriting alphanumeric characters, which contains 10 digits, 26 uppercase letters, and 26 lowercase letters, totally 62 classes. We split 15% data into test dataset. We train one model federally by using federated averaging. We utilize the popular ResNet-18 [28] structure as the architecture and add dropouts.

Traffic Prediction Traffic prediction is a crucial component in intelligent transportation systems, which has been studied for decades. In a traffic prediction task, we predict traffic in a given area and time range. The traffic datasets METR-LA and PEMS-BAY both contain traffic speed data and the corresponding timestamp on a wide range of monitored locations. The traffic readings are aggregated into 5-minute time windows. Recent advances in traffic prediction make use of convolutional neural networks and recurrent neural networks to model the spatiotemporal traffic data. In our reference implementation, we utilize the Diffusion Convolutional Recurrent Neural Network (DCRNN) [36] as the base model and apply federated averaging.

Trend Prediction Trend prediction aims to establish a model and predict future trends. To achieve this, we use the features from different datasets as the common features, and use the scores as labels. For example, for the game datasets, we can use the price, release_date, required_age, is_multiplayer columns from the Steam Game dataset, combined with the platform, genre columns from the IGN Rating datasets as features, and use the score column from any datasets as the label. Similarly, For the two songs datasets, audio features, lyrics and other columns can be used as features, and song_hotness as the label. Last, for the movie datasets, features can be tag, genre, budget, etc., and a rating column from any of the three datasets can be used as the label. To split the train and test set, we divide the aligned datasets by a time point, and use entries before that time point to train the model, and use entries after that time point to perform testing.

References

[1] John7obed/ign_games_of_20_years. https://github.com/john7obed/ign_games_of_20_years, February 2017.

[2] FederatedAI/FATE. https://github.com/FederatedAI/FATE, June 2020.

[3] OpenMined/PySyft: A library for encrypted, privacy preserving machine learning. https://github.com/OpenMined/PySyft, June 2020.

[4] PaddlePaddle/PaddleFL. https://github.com/PaddlePaddle/PaddleFL, June 2020.

[5] Tensorflow/federated: A framework for implementing federated learning. https://github.com/tensorflow/federated, June 2020.

[6] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, pages 308–318, 2016. doi: 10.1145/2976749.2978318.

[7] Eirikur Agustsson, Radu Timofte, Sergio Escalera, Xavier Baro, Isabelle Guyon, and Rasmus Rothe. Apparent and Real Age Estimation in Still Images with Deep Residual Regressors on Appa-Real Database. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 87–94, Washington, DC, DC, USA, May 2017. IEEE. ISBN 978-1-5090-4023-0. doi: 10.1109/FG.2017.20.

[8] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[9] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A Benchmark for Federated Settings. *arXiv:1812.01097 [cs, stat]*, December 2018.

[10] Jingchun Cheng, Yali Li, Jilong Wang, Le Yu, and Shengjin Wang. Exploiting effective facial patches for robust gender recognition. *Tsinghua Science and Technology*, 24(3):333–345, 2019.

[11] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. SecureBoost: A Lossless Federated Learning Framework. *arXiv:1901.08755 [cs, stat]*, January 2019.

[12] Albert Clapés, Ozan Bilici, Dariia Temirova, Egils Avots, Gholamreza Anbarjafari, and Sergio Escalera. From apparent to real age: Gender, age, ethnic, makeup, and expression bias analysis in real age estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2373–2382, 2018.

[13] Cody Coleman, Deepak Narayanan, Daniel Kang, Tian Zhao, Jian Zhang, Luigi Nardi, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102, 2017.

[14] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.

[15] Teófilo Emídio De Campos, Bodla Rakesh Babu, Manik Varma, et al. Character recognition in natural images. *VISAPP (2)*, 7, 2009.

[16] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference*, 2017.

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[18] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013. ISSN 1551-305X, 1551-3068. doi: 10.1561/0400000042.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Shai Halevi, and Tal Rabin, editors, *Theory of Cryptography*, volume 3876, pages 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-32731-8 978-3-540-32732-5. doi: 10.1007/11681878_14.

[20] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and Differential Privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, Las Vegas, NV, USA, October 2010. IEEE. ISBN 978-1-4244-8525-3. doi: 10.1109/FOCS.2010.12.

[21] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and Gender Estimation of Unfiltered Faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, December 2014. ISSN 1556-6013, 1556-6021. doi: 10.1109/TIFS.2014.2359646.

[22] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing - STOC '09*, page 169, Bethesda, MD, USA, 2009. ACM Press. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536440.

[23] Daniel Grijalva. Danielgrijalva/movie-stats. https://github.com/danielgrijalva/movie-stats, March 2020.

[24] Hu Han, Charles Otto, and Anil K. Jain. Age estimation from face images: Human vs. machine performance. In *2013 International Conference on Biometrics (ICB)*, pages 1–8, Madrid, Spain, June 2013. IEEE. ISBN 978-1-4799-0310-8. doi: 10.1109/ICB.2013.6613022.

[25] Tianshu Hao, Yunyou Huang, Xu Wen, Wanling Gao, Fan Zhang, Chen Zheng, Lei Wang, Hainan Ye, Kai Hwang, Zujie Ren, and Jianfeng Zhan. Edge AIBench: Towards Comprehensive End-to-end Edge Computing Benchmarking. *arXiv:1908.01924 [cs]*, August 2019.

[26] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv:1711.10677 [cs]*, November 2017.

[27] F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, January 2016. ISSN 2160-6455, 2160-6463. doi: 10.1145/2827872.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural Collaborative Filtering. *arXiv:1708.05031 [cs]*, August 2017.

[30] Gary B. Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007.

[31] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[32] H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, 2014. URL `http://dblp.uni-trier.de/db/journals/cacm/cacm57.html#JagadishGLPPRS14`.

[33] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *arXiv:1912.04977 [cs, stat]*, December 2019.

[34] Qinbin Li, Zeyi Wen, and Bingsheng He. Practical Federated Gradient Boosting Decision Trees. *arXiv:1911.04206 [cs, stat]*, December 2019.

[35] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, and Bingsheng He. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *arXiv:1907.09693 [cs, stat]*, December 2019.

[36] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

[37] Lifeng Liu, Fengda Zhang, Jun Xiao, and Chao Wu. Evaluation Framework For Large-scale Federated Learning. *arXiv:2003.01575 [cs, stat]*, March 2020.

[38] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[39] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. Real-World Image Datasets for Federated Learning. *arXiv:1910.11089 [cs, stat]*, October 2019.

[40] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[41] Peter Mattson, Christine Cheng, Cody Coleman, Greg Diamos, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bittorf, David Brooks, Dehao Chen, Debojyoti Dutta, Udit Gupta, Kim Hazelwood, Andrew Hock, Xinyuan Huang, Atsushi Ike, Bill Jia, Daniel Kang, David Kanter, Naveen Kumar, Jeffery Liao, Guokai Ma, Deepak Narayanan, Tayo Oguntebi, Gennady Pekhimenko, Lillian Pentecost, Vijay Janapa Reddi, Taylor Robie, Tom St John, Tsuguchika Tabaru, Carole-Jean Wu, Lingjie Xu, Masafumi Yamazaki, Cliff Young, and Matei Zaharia. MLPerf Training Benchmark. *arXiv:1910.01500 [cs, stat]*, March 2020.

[42] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd International Conference on World Wide Web - WWW '13*, pages 897–908, Rio de Janeiro, Brazil, 2013. ACM Press. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488466.

[43] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, Fort Lauderdale, FL, USA, April 2017. PMLR.

[44] Ilya Mironov. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, Santa Barbara, CA, August 2017. IEEE. ISBN 978-1-5386-3217-8. doi: 10.1109/CSF.2017.11.

[45] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft. Privacy-Preserving Ridge Regression on Hundreds of Millions of Records. In *2013 IEEE Symposium on Security and Privacy*, pages 334–348, Berkeley, CA, May 2013. IEEE. ISBN 978-0-7695-4977-4 978-1-4673-6166-8. doi: 10.1109/SP.2013.30.

[46] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A Performance Evaluation of Federated Learning Algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning - DIDL '18*, pages 1–8, Rennes, France, 2018. ACM Press. ISBN 978-1-4503-6119-4. doi: 10.1145/3286490.3286559.

[47] Mark O'Neill, Elham Vaziripour, Justin Wu, and Daniel Zappala. Condensing Steam: Distilling the Diversity of Gamer Behavior. In *Proceedings of the 2016 ACM on Internet Measurement Conference - IMC '16*, pages 81–95, Santa Monica, California, USA, 2016. ACM Press. ISBN 978-1-4503-4526-2. doi: 10.1145/2987443.2987489.

[48] Weike Pan. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, 177:447–453, 2016.

[49] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, Ramesh Chukka, Cody Coleman, Sam Davis, Pan Deng, Greg Diamos, Jared Duke, Dave Fick, J. Scott Gardner, Itay Hubara, Sachin Idgunji, Thomas B. Jablin, Jeff Jiao, Tom St John, Pankaj Kanwar, David Lee, Jeffery Liao, Anton Lokhmotov, Francisco Massa, Peng Meng, Paulius Micikevicius, Colin Osborne, Gennady Pekhimenko, Arun Tejusve Raghunath Rajan, Dilip Sequeira, Ashish Sirasao, Fei Sun, Hanlin Tang, Michael Thomson, Frank Wei, Ephrem Wu, Lingjie Xu, Koichi Yamada, Bing Yu, George Yuan, Aaron Zhong, Peizhao Zhang, and Yuchen Zhou. MLPerf Inference Benchmark. *arXiv:1911.02549 [cs, stat]*, November 2019.

[50] Rasmus Rothe, Radu Timofte, and Luc Van Gool. DEX: Deep EXpectation of Apparent Age from a Single Image. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 252–257, Santiago, Chile, December 2015. IEEE. ISBN 978-1-4673-9711-7. doi: 10.1109/ICCVW.2015.41.

[51] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018.

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[53] D. Wang, Q. Wang, C. Liu, and F. Yin. Casia online and offline chinese handwriting databases. In *2013 12th International Conference on Document Analysis and Recognition*, pages 37–41, Los Alamitos, CA, USA, sep 2011. IEEE Computer Society. doi: 10.1109/ICDAR.2011.17. URL `https://doi.ieeecomputersociety.org/10.1109/ICDAR.2011.17`.

[54] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated Learning with Matched Averaging. *arXiv:2002.06440 [cs, stat]*, February 2020.

[55] Luping Wang, Wei Wang, and Bo Li. CMFL: Mitigating Communication Overhead for Federated Learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 954–964, Dallas, TX, USA, July 2019. IEEE. ISBN 978-1-72812-519-0. doi: 10.1109/ICDCS.2019.00099.

[56] Mei Wang and Weihong Deng. Deep face recognition: A survey. *arXiv preprint arXiv:1804.06655*, 2018.

[57] Mei Wang and Weihong Deng. Mitigate bias in face recognition using skewness-aware reinforcement learning. *arXiv preprint arXiv:1911.10692*, 2019.

[58] Mei Wang, Weihong Deng, Jiani Hu, Xunqiang Tao, and Yaohai Huang. Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[59] Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1343–1353, Beijing, China, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1130.

[60] Xuefeng Xiao, Lianwen Jin, Yafeng Yang, Weixin Yang, Jun Sun, and Tianhai Chang. Building fast and compact convolutional neural networks for offline handwritten chinese character recognition, 2017.

[61] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, January 2019. ISSN 21576904. doi: 10.1145/3298981.

[62] Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982)*, pages 160–164, Chicago, IL, USA, November 1982. IEEE. doi: 10.1109/SFCS.1982.38.

[63] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[64] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), July 2018. ISSN 1942-4787, 1942-4795. doi: 10.1002/widm.1253.

[65] Yutong Zheng, Dipan K Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5089–5097, 2018.

[66] Zhuoyao Zhong, Lianwen Jin, and Zecheng Xie. High performance offline handwritten chinese character recognition using googlenet and directional feature maps, 2015.

[67] Shusen Zhou, Qingcai Chen, and Xiaolong Wang. HIT-OR3C: An opening recognition corpus for Chinese characters. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems - DAS '10*, pages 223–230, Boston, Massachusetts, 2010. ACM Press. ISBN 978-1-60558-773-8. doi: 10.1145/1815330.1815359.