A Introduction of FedScale Datasets

Category	Name	Data Type	#Clients	#Instances	Example Task
CV	iNature [5]	Image	2,295	193K	Classification
	FEMNIST [18]	Image	3,400	640K	Classification
	OpenImage [4]	Image	13,771	1.3M	Classification, Object detection
	Google Landmark [53]	Image	43,484	3.6M	Classification
	Charades [49]	Video	266	10K	Action recognition
	VLOG [23]	Video	4,900	9.6K	Classification, Object detection
	Waymo Motion [21]	Video	496,358	32.5M	Motion prediction
NLP	Europarl [35]	Text	27,835	1.2M	Text translation
	Blog Corpus [48]	Text	19,320	137M	Word prediction
	Stackoverflow [10]	Text	342,477	135M	Word prediction, Classification
	Reddit [9]	Text	1,660,820	351M	Word prediction
	Amazon Review [41]	Text	1,822,925	166M	Classification, Word prediction
	CoQA [45]	Text	7,189	114K	Question Answering
	LibriTTS [56]	Text	2,456	37K	Text to speech
	Google Speech [52]	Audio	2,618	105K	Speech recognition
	Common Voice [2]	Audio	12,976	1.1M	Speech recognition
Misc ML	Taobao [11]	Text	182,806	20.9M	Recommendation
	Fox Go [3]	Text	150,333	4.9M	Reinforcement learning

Table 4: Statistics of FedScale datasets. FedScale has 18 realistic client datasets, which are from the real-world collection, and we partitioned each dataset using its real client-data mapping.

FedScale currently has 18 realistic federated datasets across a wide range of scales and task categories (Table 4). Here, we provide the description of some representative datasets, and the reader can refer to FedScale repository (https://github.com/SymbioticLab/FedScale) for more datasets.

Google Speech Commands. A speech recognition dataset [52] with over ten thousand clips of one-second-long duration. Each clip contains one of the 35 common words (e.g., digits zero to nine, "Yes", "No", "Up", "Down") spoken by thousands of different people.

OpenImage. OpenImage [4] is a vision dataset collected from Flickr, an image and video hosting service. It contains a total of 16M bounding boxes for 600 object classes (e.g., Microwave oven). We clean up the dataset according to the provided indices of clients. In our evaluation, the size of each image is 256×256.

Reddit and StackOverflow. Reddit [9] (StackOverflow [10]) consists of comments from the Reddit (StackOverflow) website. It has been widely used for language modeling tasks, and we consider each user as a client. In this dataset, we restrict to the 30k most frequently used words, and represent each sentence as a sequence of indices corresponding to these 30k frequently used words.

VLOG. VLOG [23] is a video dataset collected from YouTube. It contains more than 10k Lifestyle Vlogs, videos that people purportedly record to show their lives, from more than 4k actors. This dataset aimed at understanding everyday human interaction and contains labels for scene classification, hand-state prediction task, and hand detection.

LibriTTS. LibriTTS [56] is a large-scale text-to-speech dataset. It is derived from audiobooks that are part of the LibriVox project [6]. There are 585 hours of read English speech from 2456 speakers at 24kHz sampling rate.

Taobao. Taobao Dataset [11] is a dataset of click rate prediction about display Ad, which is displayed on the website of Taobao. It is composed of 1,140,000 users ad display/click logs for

```
from fedscale.core.client_manager import ClientManager
import Oort

class Customized_ClientManager(ClientManager):
  def __init__(self, *args):
     super().__init__(*args)
     self.oort_selector = Oort.create_training_selector(*args)

 # Replace default client selection algorithm w/ Oort
 def resampleClients(self, numOfClients, cur_time, feedbacks):
     # Feed Oort w/ execution feedbacks from last training round
     oort_selector.update_client_info(feedbacks)
     selected_clients = oort_selector.select_participants(numOfClients, cur_time)

     return selected_clients
```

Figure 15: Evaluate new client selection algorithm [36].

```
from fedscale.core.client import Client

class Customized_Client(Client):
# Customize the training on each client
  def train(self,client_data,model,conf):
     # Get the training result from
     # the default training component
     training_result = super().train(
         client_data, model, conf)

     # Implementation of compression
     compressed_result = compress_impl(
             training_result)
     return compressed_result
```

```
from fedscale.core.client import Client

class Customized_Client(Client):
# Customize the training on each client
  def train(self,client_data,model,conf):
     # Get the training result from
     # the default training component
     training_result = super().train(
         client_data, model, conf)

     # Clip updates and add noise
     secure_result = secure_impl(
             training_result)
     return secure_result
```

Figure 16: Evaluate model compression [46].   Figure 17: Evaluate security enhancement [50].

8 days, which are randomly sampled from the website of Taobao. We partitioned it using its real client-data mapping.

Waymo Motion.   Waymo Motion [21] is composed of 103,354 segments each containing 20 seconds of object tracks at 10Hz and map data for the area covered by the segment. These segments are further broken into 9 second scenarios (8 seconds of future data and 1 second of history) with 5 second overlap, and we consider each scenario as a client.

B   Examples of New Plugins

In this section, we demonstrate several examples to show the ease of integrating today's FL efforts for realistic evaluations in FedScale.

At its core, FAR provides flexible APIs on each module so that the developer can access and customize methods of the base class. Note that FAR will automatically integrate new plugins into evaluations, and then produces practical FL metrics. Figure 15 demonstrates that we can easily evaluate new client selection algorithms, Oort [36], by modifying a few lines of the clientManager module. Similarly, Figure 16 and Figure 17 show that we can extend the basic Client module to apply new gradient compression [46] and enhancement for malicious attack [50], respectively.