HKUST SPD - INSTITUTIONAL REPOSITORY

Title	Inspecting the Running Process of Horizontal Federated Learning via Visual Analytics
Authors	Li, Quan; Wei, Xiguang; Lin, Huanbin; Liu, Yang; Chen, Tianjian; Ma, Xiaojuan
Source	IEEE Transactions on Visualization and Computer Graphics, 19 April 2021, article number 9408377
Version	Accepted Version
DOI	10.1109/tvcg.2021.3074010
Publisher	IEEE
Copyright	© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creatin

This version is available at HKUST SPD - Institutional Repository (<https://repository.ust.hk/ir>)

If it is the author's pre-published version, changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published version.

Inspecting the Running Process of Horizontal Federated Learning via Visual Analytics

Quan Li, Xiguang Wei, Huanbin Lin, Yang Liu, Tianjian Chen, and Xiaojuan Ma

Abstract—As a decentralized training approach, horizontal federated learning (HFL) enables distributed clients to collaboratively learn a machine learning model while keeping personal/private information on local devices. Despite the enhanced performance and efficiency of HFL over local training, clues for inspecting the behaviors of the participating clients and the federated model are usually lacking due to the privacy-preserving nature of HFL. Consequently, the users can only conduct a shallow-level analysis of potential abnormal behaviors and have limited means to assess the contributions of individual clients and implement the necessary intervention. Visualization techniques have been introduced to facilitate the HFL process inspection, usually by providing model metrics and evaluation results as a dashboard representation. Although the existing visualization methods allow a simple examination of the HFL model performance, they cannot support the intensive exploration of the HFL process. In this study, strictly following the HFL privacy-preserving protocol, we design an exploratory visual analytics system for the HFL process termed *HFLens*, which supports comparative visual interpretation at the overview, communication round, and client instance levels. Specifically, the proposed system facilitates the investigation of the overall process involving all clients, the correlation analysis of clients' information in one or different communication round(s), the identification of potential anomalies, and the contribution assessment of each HFL client. Two case studies confirm the efficacy of our system. Experts' feedback suggests that our approach indeed helps in understanding and diagnosing the HFL process better.

Index Terms—Federated learning, anomaly detection, contribution assessment, visualization

1 INTRODUCTION

TRADITIONAL machine learning adopts a centralized approach, which requires training data from different sources to be aggregated on a single machine or in a data center. This centralized training approach can be privacy-intrusive, requiring users to upload their data to train a better machine learning model. As an alternative decentralized training approach, federated learning (FL) enables users to collaboratively learn a machine learning model while keeping all the data that may contain private information on their local devices [1]. In this case, users can benefit from a well-trained machine learning model, utilizing sufficient data from different sources without sharing sensitive personal data.

Yang et al. [2] abstracted FL into three categories according to different application scenarios, namely, horizontal FL (HFL), vertical FL (VFL), and federated transfer learning (FTL). In this study, we focus on HFL [1], [2], [3], [4], which is also known as sample-based FL and can be applied in scenarios where datasets share the same feature space but differ in sample space (Figure 1). In other words, the datasets of different parties have the same format but collected from different sources. HFL aims to help multiple parties (e.g., data owners) who may own insufficient data with the same feature space (e.g., labeled image data) to jointly train a federated model for each to benefit from this sufficiently trained model. The word “horizontal” comes from the term “horizontal partition,” which is widely used in the context of the traditional tabular view of a database (i.e., the rows of a table are horizontally partitioned into different groups, and each row contains a complete

set of data features). Under the premise of HFL, the data collected and stored by each party need not be uploaded to a common server to facilitate model training. Instead, an initial model is sent from the HFL server to each party, that is, the client, who then uses locally stored data to refine this model. After several epochs of local training, the updated model parameters from each party are encrypted and sent back to the server. The parameters are then aggregated into a global federated model, which will be eventually distributed to the parties in the federation. This entire process is called a *communication round*. The training continues until the model converges.

Although HFL has demonstrated promising performance in industrial [5], [6], [7], medical [8], [9], [10], [11], [12], and financial [2] scenarios, real-world practitioners encounter the following challenges when trying to promote HFL in their application domains. **(1) Limited Information for Inspection.** Unlike traditional centralized machine learning in which the data center or learning server knows nearly everything about the entire system, the HFL server has neither access to the clients' data nor full control of the clients' behaviors due to its data privacy mechanism design [2]. Therefore, potential risks, such as the deliberate embedding of malicious information in clients' data, are invisible to the HFL server and may cause unexpected results in the federated system. Under this premise, failing to clearly define what information to collect and make good sense of the available information may leave federated participants, such as collaboration partners and customers, especially those with limited background in FL, very confused, potentially preventing the wide adoption of FL techniques. **(2) Shallow-level Analysis.** Tools, such as Turbofan Tycoon [13] and FATE-Board [14], have been developed to communicate the advantages of federated models. By summarizing the processing logs and the model performances, these tools can facilitate the monitoring of the execution of the federated model. However,

- Quan Li is with School of Information Science and Technology, ShanghaiTech University.
- Xiguang Wei is with AI Department, Shenzhen Semacare Medical Technology Co., Ltd.
- Huanbin Lin, Yang Liu, and Tianjian Chen are with AI Group, WeBank.
- Xiaojuan Ma is with Department of Computer Science and Engineering, The Hong Kong University of Science and Technology.

Fig. 1. The two subfigures ((a) and (b)) illustrate the structure and data of a typical HFL system. The subfigure (b) is called HFL, or sample-based federated learning, which is introduced in the scenarios where data sets share the same features but differ in samples. All the data sets have the same label. For example, two regional banks ((c) Bank A and (d) Bank B) may have different user groups (U1 – U7 and U8 – U14) from their respective regions, and the intersection set of their users is very small. However, their business is very similar, so the feature spaces (X1 and X2) are the same. The objective is that both banks want to jointly build an anti-money laundering model. There are a few assumptions in this scenario: (1) Label Y indicates whether the corresponding sample has the money laundering; (2) Both parties have the same features (X1, X2) and label (Y); and (3) Neither party wants to expose its own data for jointly training the model. The challenge is that both parties do not have enough data entries for training a good enough model. By leveraging HFL, the expected result is to establish a joint model while protecting privacy and the performance of the joint model is better than the unilateral data model. For more information regarding HFL, please refer to [2].

fine-grained analysis, such as potential anomaly detection and contribution assessment, is challenging for the existing tools. First, during the model training process of HFL, many basic operations, such as controlling the quality of data samples and identifying whether some clients are abnormal, are hindered by the privacy-preserving mechanism design of HFL. Without effective analysis and support for the subsequent adjustments for optimization, the efficacy of the entire model training would be affected. Second, in practice, multiple institutions may initially form a HFL framework because of their lack of data. Maintaining the federation can be difficult if the contributions of clients to the federated model over time cannot be assessed. The limited information must be exploited to make an effective judgment of potential abnormal clients/events and formulate a reasonable contribution measurement scheme. (3) **Inefficient Intervention.** Traditional centralized machine learning typically trains and infers models separately (i.e., a model is trained first, and then the possible issues of the model are inferred), whereas HFL systems usually couple the training and inference processes. That is, HFL can be a continuous process that continuously updates the model to adapt to possible data distribution changes. Despite this advantage, in practice, HFL maintainers have trouble making swift and informed model-adapting decisions in a short period based only on some simple performance logs and metrics. Thus,

the effective representation of the involved information of the running process from different clients in the federation over time is urgently needed. Achieving this can facilitate a phased adjustment of the strategy for the passing communication rounds. That is, a timely review of the HFL process can provide effective intervention for future HFL execution. Furthermore, current studies on HFL mainly focus on issues, like optimization methods [15], [16], [17], [18], privacy protection [12], and theoretical outlier detection [19], [20], and few studies focus on inspecting the running process. Empirical studies on the extent to which a visualization system can resolve breakdowns by increasing the accessibility of the HFL process and result and bridging the HFL practitioners' knowledge with HFL are limited.

In this study, we help HFL practitioners understand and inspect the HFL process. We first conduct an observational study of our collaboration experts' current practices to identify their primary needs and concerns regarding the use of HFL. We then streamline the analytical pipeline and set up experiments to collect the general information of a typical HFL process. According to the findings and the information, we propose an interactive visual analytics approach named *HFLens* to help domain experts maintain an overview of the HFL execution process, access the status of each communication round and client, identify possible anomalies, and evaluate the contribution of each client. *HFLens* supports fine-grained analysis at the overview, communication round, and client levels. Two case studies and experts' feedback confirm the system's efficacy. Our primary contributions are as follows.

- We characterize the issues in the context of the HFL process through an observational study and a thorough discussion of the design requirements with HFL experts.
- We develop interactive visualizations enhanced with new visual features to support the fine-grained analysis of the HFL process at the overview, communication round, and client instance levels. To the best of our knowledge, *HFLens* is the first endeavor of this kind.

2 RELATED WORK

The literature that overlaps this work can be categorized into four groups: 1) FL; 2) FL visualization; 3) interpretable machine learning, and 4) anomaly detection and contribution assessment in FL.

2.1 Federated Learning

FL was first proposed by Google, which trains machine learning models using the decentralized data on end devices, such as mobile phones [1]. The first commercial FL application is a keyboard app on a mobile phone called GBoard [21] in which a recurrent neural network language model is trained using FL for next-word prediction in a virtual keyboard. GBoard's authors also built a scalable production system for FL in the domain of mobile devices [22] based on TensorFlow and high-level design, which has been deployed in production and solves applied learning problems over tens of millions of real-world devices. These works are classified as HFL because the clients share the same feature domain but have different samples. Many FL applications currently exist, covering industrial [5], [6], [7], medical [8], [9], [10], [11], [12], and financial [2] scenarios.

As previously mentioned, FL can be categorized into three groups: HFL, VFL, and FTL [2]. HFL applies to circumstances where many overlaps on features exist but only a few on instances,

such as the Google GBoard use case in which models can be aggregated directly from the edge models. VFL applies to scenarios where many overlapping instances but few overlapping features exist [23], [24]. An example is the relationship of insurers and online retailers. Both have many overlapping users, but each have its own feature space and labels. VFL merges the features to create a larger feature space for machine learning tasks and uses homomorphic encryption to provide data privacy protection for the involved parties. FTL uses transfer learning [25] to improve model performance when few overlaps on features and instances exist [26]. In this paper, we focus on HFL, which is a decentralized learning approach that uses a secure aggregation scheme to protect the privacy of users.

2.2 Federated Learning Visualization

Lenovo demonstrates HFL by simulating an industrial process in factories [27] to predict the pressure inside the hardware based on HFL. Cloudera Fast Forward Labs showcases an interactive simulation prototype, *Turbofan Tycoon*, that leverages visualizations to communicate the advantages of the federated model that makes accurate predictions about when a turbofan will fail [13]. The two studies aim to intuitively show how HFL works but mainly focus on the performance improvement brought by HFL instead of inspecting the HFL process. *FATEBoard* is a visualization component in *FATE*¹, which is an end-to-end pipeline platform for VFL [14]. *FATEBoard* uses a dashboard to show modeling log metrics and evaluation results, including the *dataset info*, *job status*, *graph*, and *model output* components. Thus, *FATEBoard* records and summarizes the logs in the VFL process but cannot provide detailed analysis when anomalies exist among the clients. Wei et al. [28] developed a game to demonstrate the HFL performance and further built a visualization prototype system to show the HFL process. This prototype visualizes the client data, monitors the aggregated accuracy at the server-side, and shows potential factors regarding the contribution of HFL clients by using a Radar chart and the ranking of all the participating clients. However, this work is highly customized for the car racing game and can be rather difficult to be generalized for other scenarios. Furthermore, in a real HFL system, the data of the clients are private and cannot be “seen” by the server-side. Our work strictly follows the data privacy-preserving design of the HFL system, that is, the private data from the client-side cannot be accessed.

2.3 Interpretable Machine Learning

In practice, interpretability is a desirable property of machine learning models in critical areas, such as security, medicine, and finance [29]. For example, a quantitative trader may prefer an interpretable model with low expected return due to its predictability and low risk. Unfortunately, the best-performing models (e.g., deep neural networks) in many applications are complex models whose predictions are difficult to explain. Thus, growing interest to understand, diagnose, and explain intelligent systems exists in both academia and industry [29]. The efforts towards more explainable models are categorized into two types: 1) designing models with simple structures that can be easily understood or models that are interpretable on a modular level, and 2) developing methods that generate explanations for machine learning models (already complicated models). Linear regression [30], logistic

regression [31], and the decision tree [30] are the commonly used interpretable models, which have good explainability because humans can easily understand the model’s reasoning by simply looking at the coefficients of each feature. Methods that generate explanations are also called model-agnostic methods, such as partial dependence plot (PDP) [32], Shapley values [33], and individual conditional expectation (ICE) [34]. For example, Wang [35] proposed a method for balancing the model interpretability and data privacy in VFL by using Shapley values to reveal the detailed feature importance for host features and a unified importance value for federated guest features. The great advantage of model-agnostic interpretation methods over model-specific ones is their flexibility [36]. Generating explanations of a model without modifying the model itself is preferred when the underlying model is already extremely complicated, such as neural networks [37]. Thus, the key issue of explainability is the cognitive capability of a human.

Model explanations are human-understandable representations that identify the causes of the model’s prediction(s). A typical form of human-understandable representations is visualization. Visualization and visual analytics systems are semi-automatic solutions at the **different life cycle stages** (i.e., data engineering, model development, and model operation) of an intelligent system for an explainable model [29]. In the data engineering stage, data visualization can help humans explore the data and obtain a qualitative sense of the nature of the data [38], [39]. In the model development stage, visual analytics systems serve as development tools, which increase the transparency and understandability of the development [40], [41], [42], [43], [44], [45]. At the last stage, that is, model operation, visualization can help explain the inner workings of the system to end-users during model deployment. However, this stage has received little attention. Ideally, two common tasks that need explainability are identified: trust establishment and inspection/monitoring. Gaining users’ trust is the first challenge after model deployment. Ribeiro et al. [46] designed an inspiring solution for explanation, that is, locally approximating the complex model using an interpretable one, a linear model, which is used to generate a local explanation. After establishing the trust of the end-users toward a ML model, a human-friendly interface is needed for inspecting or monitoring the model. In this work, we focus on inspecting the running process in HFL, which belongs to the visualization role applied in the model operation stage. To the best of our knowledge, we are the first to attempt such an endeavor in HFL.

2.4 Anomaly Detection and Contribution Assessment

Identifying possible anomalies and studying the incentive mechanism in HFL are increasingly important research topics due to the high demand of the HFL eco-system. Studies state that abnormal clients may have a differentiable pattern in the model and gradient histograms [2], [47]. An adversary lens was proposed to analyze HFL by attacking the HFL system using model poison [20]. Li et al. [19] leveraged an auto-encoder-decoder to encode entire/partial model parameters and used reconstruction error as a possible score to index anomalies. However, this approach requires a significantly large number of clients. Inspired by the above studies, we leverage model and gradient histograms, which contain little client information to help determine the potential anomalies.

Incentive mechanisms that motivate the clients to join the federation learning process have also been largely overlooked.

1. <https://fate.fedai.org/>

Without a well-designed incentive mechanism, clients would be unwilling to join HFL due to the considerable overhead in terms of computation and communication costs during the HFL process. Studies indicate that the sampling count on each client and the local model performance may reflect the data quality [1], [2] and can therefore be roughly used to evaluate clients' contributions. Wang et al. [48] calculated the contributions in the context of both HFL and VFL. For HFL, they used the deletion method to calculate client influence. Kang et al. [49] adopted the contract theory and designed an incentive mechanism in HFL. However, these two methods require running the entire HFL process multiple times until a score is assigned to each client. In this study, we visually summarize all the above-mentioned factors and leverage pairwise comparison to assess the contribution of the clients by ranking.

3 OBSERVATIONAL STUDY AND REQUIREMENTS

3.1 Conventional Practice and Bottlenecks

In typical commercial scenarios, the following anecdotal conversation between the customer side (C) and the AI solution provider (P) can often be heard [50]:

C: *"We need a solution for detecting flames in a factory floor from surveillance videos to provide early warning of fire."*

P: *"No problem. We will need some data to train a flame detection model. How much data with annotations do you have?"*

C: *"We just launched our system so we haven't got enough data with annotations yet."*

P: *"Then, how about we unite other customers with similar needs? Each of you contributes a portion of data so we can have enough data to train the model."*

C: *"That's cool but our data contain sensitive information, we cannot pass them on to a third party."*

In a centralized machine learning approach, data privacy-preserving regulations are impossible to maintain. Furthermore, the entire process of model updating usually in a centralized machine learning approach usually takes a long time, depending on when the next round of off-line training occurs. In other words, when users accumulate new data, they must upload the data to the central training server and wait for the next round of training, which leads to the problem of lagging feedback from clients to the machine learning server. As a data privacy-preserving strategy, HFL enables the local training of a visual object detection model at the data owner's site and the uploading of model parameters to a central server for aggregation instead of uploading the actual training dataset.

To understand how (well) HFL is used in practice, we worked with a team of experts from a collaborating local AI solution provider, including one product manager (E.1), two HFL researchers (E.2 – E.3), and one HFL framework developer and maintainer (E.4). A considerable part of their job is to provide AI solutions of computer vision to their customers with the same need. They jointly launched an object detection application, that is, detecting flames in a factory floor from surveillance videos by using HFL. Specifically, during the running process of the HFL system, E.4 would inspect the HFL system every five communication rounds to 1) identify potential anomalies in model training, 2) inspect model issues and 3) assess the contribution of each client in the federated system based on the collected logs. If anomalies are found, the experts would 1) block the models from those anomalous clients to be aggregated to the federated model, 2) adjust the weighting strategy for aggregation, or 3) recover a

previously aggregated federated model in an earlier communication round.

During the inspection process, the experts encountered several issues. First, manually compiling logs across several communication rounds and then separately inspecting each metric are tedious and labor-intensive. For example, when the performance of the federated model drops, they must identify which clients may be responsible for the declined model performance by inspecting each client. Second, the current practice does not support the correlation analysis of relevant information, which may lead to a biased conclusion. Third, when assessing the contribution of each client, no means can be adopted to compile all available metrics and parameters for an objective and comprehensive judgment.

3.2 Experts' Needs and Expectations

To ensure that the ontological structure of our approach fits the domain tasks well, we interviewed the experts (E.1 – E.4) in four separate sessions to identify their primary concerns about the use of HFL and the potential obstacles to efficiently obtaining knowledge about the HFL process. At the end of the interviews, the need for a visualization system to ground the team's conversation with HFL emerged as a key theme among the feedback collected. Despite differences in individual expectations for such a system, certain requirements were expressed across the board.

R.1 Understand the HFL process. According to E.1 – E.3, the clients' status may vary during the HFL. For example, a client may join the federated learning process from the very beginning and leave in the next communication round due to network connection errors or manual interruption issues. Thus, maintaining a clear overview of the status of all clients, such as whether they are participating in the current communication round, the sample counts of local datasets, and the model performance before and after a certain communication round, is critical.

R.2 Correlate client information in one/different round(s). The experts (E.2 – E.3) commented that one pressing issue of a traditional FL dashboard, like *FATEBoard* [14], is its way of presenting modeling metrics and evaluation results as separate logs. Given that the spatiotemporally distributed HFL data of clients are correlated in one or different round(s), displaying HFL client data as individual metrics without considering the correlation among them would hinder the observation of the evolution of the HFL process.

R.3 Inspect "clusters" and "outliers" of clients. During the running process of HFL, the evolving federated model aggregated by the server is sent back and forth between the clients and the server. E.4 frequently raised that some clients exhibit similar performance while others present different model accuracies and losses. Therefore, he is curious about his capability of collectively identifying those clients reporting a similar performance and inspecting those reporting a different performance across different communication rounds.

R.4 Observe potential anomalies. Given the data privacy-preserving nature of HFL, limited information can be attained from the HFL server compared with that generated by centralized machine learning systems. The experts found the occurrence of potential incidents of anomalous clients and their severity from such limited and unorganized information difficult to determine. Therefore, they wondered if more information could be leveraged to identify anomalous clients who may disrupt a HFL process.

R.5 Assess the contribution of clients. The study of the incentive mechanism is a key area of HFL research. Each client

needs a reason to contribute computational power and locally fine-tuned model to the federation [49]. Therefore, determining the contribution of each client objectively and comprehensively is important for designing a good incentive mechanism in HFL. Determining the data quality of each client is difficult for the federated server due to the data privacy-preserving mechanism, but E.4 hoped to assess the contributions of clients for each communication round via other relevant factors, such as the number of participating rounds and the local model performance.

4 APPROACH OVERVIEW

We draw inspiration from the observational study of the experts' conventional practices to inform the pipeline of our approach, such as providing an overview of the HFL running process and detecting potential anomalous clients. Based on the preceding requirements, we propose a visual analytics system that supports HFL practitioners to understand and inspect the HFL process for passing communication rounds. Figure 2 illustrates the proposed pipeline, which consists of the *back-end engine* module and the *front-end visualization* module. Particularly, the *back-end engine* module comprises the phases of the HFL setting and optimization that collect the log data during a HFL process. The logs are organized for preprocessing and then fed into the *front-end visualization* module for further visual analytics.

Fig. 2. An overview of our approach. The *back-end engine* module initializes, optimizes, and collects the log data during the HFL running process. The *front-end visualization* module visualizes the extracted log data and supports an interactive data analysis.

5 BACK-END ENGINE

5.1 HFL Configuration

A typical HFL architecture consists of a HFL server and multiple clients. HFL requires that all the local data stay in local devices while a machine learning model is trained collectively. HFL configurations can be different regarding different scenarios and applications [1], [10]. To unify the input and output of a typical HFL running process, we define the configuration for a HFL server and HFL client as follows:

HFL Server. We define a set of data for model performance testing as D_{test} , which will be sent to all the clients for the local model performance validation.

HFL Client. Each client k has its data that accumulates over time. In communication round r , the clients use their own data denoted by D_k^r for model fine-tuning, and the performance of the fine-tuned model P_k^r is calculated on the test data D_{test} which is from the server. Particularly, the performance of the fine-tuned model P_k^r includes the loss ($Loss_k^r$) and the accuracy (Acc_k^r).

5.2 HFL Optimization

The HFL optimization phase consists of the following steps:

- 1) **Initialization.** A global model is firstly initialized as $W^{(0)}$ and then distributed to all participated clients.
- 2) **Fine-tuning.** At round r , the global model $W^{(r)}$ in each client is scheduled to be fine-tuned using their local data. We denote the fine-tuned model of client k as $W_k^{(r+1)}$ and it is protected by encryption methods such as Differential Privacy (DP) and Homomorphic Encryption (HE) [51].
- 3) **Collection.** The HFL server then collects all the fine-tuned models $[W_k^{(r+1)}], k \in 1 \dots K$ and the auxiliary data $[A_k^{(r+1)}], k \in 1 \dots K$ from each client. Note that in the algorithm, the auxiliary data $A_k^{(r+1)}$ usually only include the local sample count and the local model performance.
- 4) **Aggregation.** The collected models are aggregated to $W^{(r+1)}$ following function

$$W^{(r+1)} \leftarrow W^{(r)} + \frac{1}{\sum_k N_k^{(r)}} \sum_k (N_k^{(r)} \cdot \Delta W_k^{(r)}) \quad (1)$$

Note that $N_k^{(r)}$ is the sample count at round r of client k while Δ indicates the updated parts of the model parameters. For example, $\Delta W_k^{(r)}$ is the gradients of model weights at round r of client k .

- 5) **Distribution.** The aggregated model is distributed to all HFL clients.
- 6) **Repeating.** Repeat step 2 – 5 until convergence.

5.3 Data Description for Visualization

TABLE 1
Information recorded in each HFL communication round.

	Feature Name	Symbol
Client	Client ID	k
	Round ID	r
	Local Sample Count	$N_k^{(r)}$
	Model Weights (Protected)	$W_k^{(r)}$
	Local Test performance	$P_k^{(r)}$
	Start Time	$t_k^{start(r)}$
	End Time	$t_k^{end(r)}$
	Model Weight Histogram	$H_k^{model(r)}$
	Model Gradient Histogram	$H_k^{gradient(r)}$
Server	Test Performance on Server	$P_{server}^{(r)}$
	Model Histogram on Server	$H_{server}^{model(r)}$
	Model Gradient Histogram on Server	$H_{server}^{gradient(r)}$

Based on the above HFL setting and running process, we summarize the data used for visualizations as shown in Table 1. Taking the data of a single client k at the r^{th} communication round for example, model weights exposed and transferred to the server is encrypted using DP and denoted as $W_k^{(r)}$. The auxiliary data $A_k^{(r)}$ of client k include all the data that can be exposed to the server, including the client ID k and the communication round r which are used to verify possible violations in HFL aggregation. The local sampling count is denoted as $N_k^{(r)}$ and $P_k^{(r)}$ records the local model performance tested on D_{test} . The starting time and ending time of local fine-tuning are defined as $t_k^{start(r)}$ and $t_k^{end(r)}$. In addition, since $W_k^{(r)}$ is encrypted and for better understanding of the client fine-tuning status and also inspired by [20] who

Fig. 3. *HFLens* interface: (A) A data loader for selecting cases of interest; (B) The summary and projection view show the statistical information of the dataset and the 2D embedding projection of all pairs of “client id/server – round id”; (C) The HFL overview presents the participating status of all the involved HFL clients during the HFL running process; (D) The pairwise comparison view displays the metric details of each HFL client in selected communication rounds; (E) The contribution ranking view shows the ranking distributions of all the involved HFL clients. Their rankings are calculated based on metrics such as *loss*, *accuracy*, *sample count*, *model similarity*, and *gradient similarity*, the same metrics used in the pairwise comparison view. (1) Click the 10th communication round and the pairwise comparison view displays the context of the corresponding communication round (i.e., from the 8th communication round to the 12th). (2) We sort the clients based on the Cosine distance between the HFL clients and the HFL server in the final communication round. (3)(4) We find that the first HFL client in the 10th communication round has a large loss and thus we click on the corresponding rectangle. The corresponding client and its continuity are highlighted by red color. (5) The corresponding accuracy, loss and participation status of the selected HFL client are highlighted and (6) the projection view shows its position (red point) in the 2D embedding space, while (7) other normal HFL clients in this communication round are distributed in other places.

indicated that the abnormal clients may demonstrate “*differentiable behaviors in the model and gradient histogram*”, in our HFL setting, we require all clients to send their histogram of model weights $H_k^{model(r)}$ and model gradients $H_k^{gradient(r)}$. Specifically, modern machine learning models or deep learning models usually contain thousands of parameters but may carry private information from data [52]. However, the histogram of the model parameters or model gradients doesn’t contain private information but could help distinguish model training directions, i.e., the local model is becoming better or worse compared with the previous one [20]. What’s more, we use the histogram of the model parameters or model gradients instead of focusing on the details such as the order of the weights and gradients because the histogram can reflect whether the model tends to be stable or abnormal from a statistical point of view. Regarding the server side, although theoretically we have a full control of all the received fine-tuned models, the models from the clients are protected by DP and we cannot evaluate each model independently. Therefore, what we could leverage are the test performance of the aggregated model $P_{server}^{(r)}$, the histogram of aggregated model weights $H_{server}^{model(r)}$ and the histogram of the aggregated model gradients $H_{server}^{gradient(r)}$.

6 FRONT-END VISUALIZATION

HFLens leverages or augments familiar visual metaphors to enable domain experts to focus on analysis [53]. We strictly

follow the mantra “*overview first, zoom and filter, then details-on-demand*,” [54] to guide users in exploring the HFL running process. Based on these principles and the preceding requirements, we develop four major visualizations that allow the output of the HFL process to be easily inspected at the overview, communication round, and client levels. Specifically, we design a HFL overview (Figure 3(C)) to illustrate the overall running process of HFL; a pairwise comparison view (Figure 3(D)) to facilitate the exploration of clients’ status; a projection view (Figure 3(B)) to show the 2D distribution of all clients and server in different communication rounds, and a contribution ranking view (Figure 3(E)) to enable the domain experts to assess the contributions of clients.

6.1 HFL Overview for Inspecting HFL Running Process

The HFL overview (Figure 3(C))(R.1) visualizes the HFL running process of all the involved clients. The input of the HFL overview includes all participated clients in all communication rounds in terms of client ID, name and the corresponding start and end time of the local fine-tuning process. Besides, the sampling count for training, the accuracy, and the loss of the local model are also recorded. These dimensions collectively characterize the dynamic participation HFL running process.

Visualize the Participation of HFL Clients. In the HFL overview (Figure 3(C)), we align each communication round in order along the y-axis. However, determining the coordinates along

Fig. 4. (a) An example of a HFL process in three subsequent communication rounds. The links with arrows indicate the communication between two parties. (b) An illustration of HFL overview design for inspecting HFL running process based on the example of (a). The links indicate whether the corresponding nodes participate in the next communication round, i.e., the continuity of participation in various rounds. Note that in real visualization, we only render the HFL client nodes in the HFL overview, i.e., we ignore the server node in the visualization.

the y-axis of each communication round is easy while placing the x-coordinates of each communication round to minimize the number of potential link crossing is an NP-hard problem and is difficult to get an exact solution. We, therefore, use a heuristics approach following the idea of narrative visualization [55]. Suppose we have a series of communication rounds and we represent them as rounded rectangles (Figure 4(b)). In each rectangle, dots indicate the clients that participate in this round. We first align the communication rounds in the y-direction according to the round id (i.e., the order of appearance). The x-coordinate of each communication round is adjusted to minimize the overall curvatures and intersections when linking identical clients across consecutive communication rounds. Particularly, the x-coordinate in one communication round lies within the x-range of the median of the clients that participate in this round. We sort the communication rounds in the x-direction to ensure that the densest ones (the one contains the most clients) are far away from each other as possible to avoid possible visual clutter, as well as maximize the utilization of the available space. We denote all the clients that participate in the HFL process as $C = c_0, \dots, c_k$ with the index of k defining the order of their appearance. We then calculate a temp x-coordinate for each client as $temp_x(c_k) = Index(c_k) * space$, in which $space$ is a constant value indicating a horizontal gap between two adjacent clients. For example, as shown in Figure 5, for communication round r_i which contains a client set as A, B , and C , we attain its x-coordinate of the communication round rectangle as

$$X(r_i) = \frac{(temp_x(A) + temp_x(B) + temp_x(C))}{3} - H(r_i)/2 \quad (2)$$

where $H(r_i)$ is the width of the rectangle of communication round r_i which is proportional to the inside client count. With $H(r_i)$ and $X(r_i)$, we layout the clients contained in r_i evenly in the range of the communication round rectangle according to the order of their appearance. After determining all the positions of communication rounds and each client in every communication round, we render the visualization of the participation of HFL clients as shown in Figure 3(C). The color of red encodes the client's status of leaving in the next communication round and blue indicates the client's status of joining in the corresponding communication round. Note that the status of joining is determined by comparing its appearance with that in the previous round and the status of leaving is determined by comparing its appearance with that in the next round. We link the identical clients across

subsequent communication rounds by curves. The participation frequency of the HFL clients is further encoded by the curve opacity. That is, more visible line indicates a higher participation frequency of the corresponding HFL client.

Fig. 5. The layout of the communication round and HFL client positions.

Visualize Changes of HFL Network. To measure the possible changes of HFL system during the running process, we consider the federation system in one communication round as a network and introduce a metric named Overall Change Ratio (OCR) based on a metric named Change Centrality (CC) [56], which measures the changing process of a network. The OCR of a HFL network between the adjacent communication rounds in a HFL process is calculated and visualized as a line chart as shown in the left side of Figure 3(C). Particularly, given a HFL network $G = (V, E, T)$, for two consecutive communication rounds r_1 and r_2 , the n -step change ratio of a client i

$$CR_{r_1, r_2}^n(i) = \frac{|N_{r_1}^n(i) \Delta N_{r_2}^n(i)|}{|N_{r_1}^n(i) \cup N_{r_2}^n(i)|} \quad (3)$$

where $N_r^n(i) = \{v : v \in V, d(v, i) = n\}$ is the set of n -step neighbors of node i in communication round r . The numerator takes the symmetric difference of the two sets and the denominator is the size of the union of the two sets. The 1-step change ratio of node i takes the minimum value of 0 when the node owns the same 1-step neighbors from communication round r_1 to r_2 , and take the maximum value of 1 when all the 1-step neighbors of the node have changed. The N -step change ratio follows the same rules. Note that in a HFL network, all the clients are a 1-step distance away from the HFL server and at a most 2-step distance from other HFL clients, so n -step can be 0, 1, and 2. In our HFL setting, we assume that the total duration for all HFL clients to finish their local fine-tuning in one communication round is fixed, which is also the default time interval between two communication rounds.

TABLE 2

Calculation of 0/1/2-step change ratios and change centralities between communication round r_1 and r_2 of the example HFL network in Figure 4(a).

HFL node i	CR_{r_1,r_2}^0	CR_{r_1,r_2}^1	CR_{r_1,r_2}^2	CC_{r_1,r_2}	CR_{r_2,r_3}^0	CR_{r_2,r_3}^1	CR_{r_2,r_3}^2	CC_{r_2,r_3}
Server	$\frac{ \emptyset }{ \{server\} } = 0$	$\frac{ \{B,C,D,E\} }{ \{A,B,C,D,E\} } = 0.8$	0	0.2	$\frac{ \emptyset }{ \{server\} } = 0$	$\frac{ \{A,B,C,E\} }{ \{A,B,C,D,E\} } = 0.8$	0	0.2
Client A	$\frac{ \emptyset }{ \{A\} } = 0$	$\frac{ \emptyset }{ \{server\} } = 0$	$\frac{ \{D,E,B,C\} }{ \{D,E,B,C\} } = 1$	0.125	$\frac{ \{A\} }{ \{A\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{D,E\} }{ \{D,E\} } = 1$	0.875
Client B	$\frac{ \{B\} }{ \{B\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{A,C\} }{ \{A,C\} } = 1$	0.875	$\frac{ \{B\} }{ \{B\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{D,C\} }{ \{D,C\} } = 1$	0.875
Client C	$\frac{ \{C\} }{ \{C\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{A,B\} }{ \{A,B\} } = 1$	0.875	$\frac{ \{C\} }{ \{C\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{B,D\} }{ \{B,D\} } = 1$	0.875
Client D	$\frac{ \{D\} }{ \{D\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{A,E\} }{ \{A,E\} } = 1$	0.875	$\frac{ \emptyset }{ \{D\} } = 0$	$\frac{ \emptyset }{ \{server\} } = 0$	$\frac{ \{A,B,C,E\} }{ \{A,B,C,E\} } = 1$	0.125
Client E	$\frac{ \{E\} }{ \{E\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{A,D\} }{ \{A,D\} } = 1$	0.875	$\frac{ \{E\} }{ \{E\} } = 1$	$\frac{ \{server\} }{ \{server\} } = 1$	$\frac{ \{A,D\} }{ \{A,D\} } = 1$	0.875

In a HFL network, the change centrality of a HFL client i between communication round r_1 and r_2 is defined as

$$CC_{r_1,r_2}(i) = \sum_{n=0}^{e_i} a_n CR_{r_1,r_2}^n(i) \quad (4)$$

where $e_i = \max_{t \in r_1, r_2} e_t(i)$ is the maximum eccentricity (i.e., the graph diameter) of the client i , and a_n is a linear coefficient. Particularly, a_n is a decreasing sequence of n , considering that larger n means a farther distance from client i , and hence has a lower influence on the centrality. Here, we use $a_n = (1/2)^{(n+1)}$ which gives CC the property that

$$CC_{r_1,r_2}(i) = \sum_{n=0}^{e_i} \frac{1}{2^{n+1}} CR_{r_1,r_2}^n(i) = \frac{1}{2} \sum_{n=0}^{e_i} \frac{1}{2^n} CR_{r_1,r_2}^n(i) < 1 \quad (5)$$

. The change centrality of a client measures the change of its connections over time, taking into account its 1- and 2-step neighbors [56]. We define the total change of a HFL network G between two consecutive communication rounds r_1 and r_2 as

$$OCR_{r_1,r_2} = \sum_{i=0}^n CC_{r_1,r_2}(i) \quad (6)$$

where n is the number of nodes in G , including the HFL server node and all HFL client nodes in the two communication rounds. This metric is very intuitive for measuring the changes (i.e., status of joining or leaving) of a HFL network. Note that we leverage the change centrality due to the following reasons. First, the well-established aggregated structural metrics lack local details and might hide certain changes [56]. For example, if we only look at the variation of the degree centrality of a certain node, we might not see any change of its relationships if the number of added links is equal to the number of deleted ones; similarly, if we look at the variation of the betweenness centrality or the closeness centrality we might not see any change if they are symmetric. Second, conversely, the highlighting of appearing or disappearing links between two nodes might not help the analyst understand the impact this change has on the connectivity of the overall structure.

For simplicity, we showcase a sample running process that lasts for three communication rounds of one HFL server and five clients to illustrate the basic idea of the HFL overview. As shown in Figure 4(a), the HFL process starts with one HFL server and three clients, i.e., client A, client B, and client C in the 1st communication round, then client B and C leave the federated system but client D and E join the running process in the 2nd communication round. In the 3rd communication round, client D stays in the federated as well

as client B and C participate in the running process again instead of client A and E. We illustrate how we obtain the changes of a HFL network between consecutive communication rounds (i.e., from r_1 to r_3) based on this illustrative example. For example, for the 0-step change ratio of the node A, we check the status of itself and identify that A exists in both round 1 and round 2. Therefore, the numerator is the size of $\emptyset$ because no nodes are added or removed, and the denominator is the size of A, i.e., 1, since A exists in both rounds; for the 0-step change ratio of the Server node, i.e., from the 1st communication round to the 2nd, the 0-step of the HFL server is the server itself, thereby, the CR_{r_1,r_2}^0 is $\frac{|\emptyset|}{|\{server\}|}$. For the 1-step change ratio of the Server node, the numerator can be considered as the number of the direct connections² added (D, E) and removed (B, C) of the client i and the denominator is the number of the direct connections added (D, E), removed (B, C), and remained (A) direct connections of the client i . That is, B, C, D, E is either newly added or removed while only A remains in both communication rounds. Therefore, according to the definition of the n -step change ratio of a client, i.e., Equation 3, CR_{r_1,r_2}^1 is $\frac{|\{B,C,D,E\}|}{|\{A,B,C,D,E\}|}$ which is 0.8. There is no 2-step neighbors for the HFL server node. The rest can be done in the same manner. Therefore, the change centrality CC_{r_1,r_2} of the HFL server is $\frac{1}{2} \times 0 + \frac{1}{4} \times 0.8 + \frac{1}{8} \times 0 = 0.2$. We list the change centrality of each HFL node as shown in Table 2, in which we observe that the HFL nodes with the largest value of change centrality are B, C, D, and E (which is present in only one of the two communication rounds (i.e., r_1, r_2)). The HFL server node has an intermediate value since it loses two HFL clients but connects two new ones. The HFL Client A is the one with the lowest change centrality, since it appears in the connection with the HFL server in both r_1 and r_2 . Thus, the total change of the example HFL network from r_1 to r_2 OCR_{r_1,r_2} is $0.2 + 0.125 + 0.875 \times 4 = 3.825$. Similarly, OCR_{r_2,r_3} is also 3.825 from r_2 to r_3 .

Visualize Auxiliary Information. Besides visualizing the participation status of the clients during the HFL running process, additional information including the accuracy and loss of the local fine-tuned model in each communication round, the sample count for local training, and the running duration of each participated client should also be displayed to facilitate domain experts in obtaining the insights. Particularly, as shown in the part of “Loss/Accuracy” of Figure 3(C), the x-axis represents the loss/accuracy value and we use a dot to represent the accuracy and loss for each client. The distributions of accuracy and loss

2. Direct connection means the nodes directly connect to the specified node.

are visualized via box plots. A blue curve links the median value across all communication rounds. In the part of “Sample Count” of Figure 3(C), the sample count of local training for all clients in each communication round is visualized as a curve area with the x -axis representing the ids of the HFL clients and the y -axis indicating the sample count of the corresponding HFL client. An additional bar beside the curve area indicates the total sample count for local training in the corresponding communication round. To visualize the running duration in each communication round, as shown in the part of “Running Duration” of Figure 3(C), we link the start and end time of each client in each communication round. We assume that the total duration for all clients to finish their local fine-tuning in one communication round is the same. Therefore, the slope of the link indicates the model convergence speed of a certain client.

6.2 Projection View for Anomaly and Cluster Analysis

The projection view facilitates the identification of the potential anomalous clients and observation of “clusters” of clients (R.3 – R.4). Dimensionality reduction techniques such as t -SNE, PCA, and MDS that create low-dimensional representations and preserve local similarities to convey neighborhood structure [57] have been widely adopted to explore or illustrate the patterns inside a high dimension space [58]. We follow this conventional practice and project the pairs of “client id/server – round id” onto a 2D space to view the potential clusters and outliers.

Fig. 6. The projection view visualizes each HFL entity as a point in a 2D space which presents a pair of “client id/server – round id”.

Metrics for Projection. After discussing with the experts, we adopt the preceding metrics (i.e., local model performance $P_k^{(r)}$ that includes the accuracy $Acc_k^{(r)}$ and loss $Loss_k^{(r)}$, the local sample count for model fine-tuning $N_k^{(r)}$, the histogram of the fine-tuned model weight $H_k^{model(r)}$ and the histogram of the model gradient $H_k^{gradient(r)}$) of client k in a specific communication round r to evaluate a client at the instance level, thus enabling domain experts to conduct a detailed comparison between two clients in terms of these metrics.

Visual Designs. After obtaining the above metrics, we can attain the corresponding feature vector for each HFL entity. The HFL entities represent all the HFL clients and server instances in different communication rounds. We then compute the pairwise similarity between two HFL entities by using Canberra Distance [59], [60]:

$$dCan(P, Q) = \sum_i^n \frac{|P_i - Q_i|}{(P_i + Q_i)} \quad (7)$$

where P and Q represent the feature vectors of two HFL entities. We choose Canberra Distance due to its insensitivity to small changes and normalization of the absolute difference of individual comparisons, benefiting to the detection of clusters and outliers [59]. Therefore, an entire distance matrix can be obtained for the following dimensionality reduction analysis. Particularly, we construct a 2D embedding of HFL entities’ metrics using t -SNE based on the attained distance matrix. We select t -SNE because it shows superiority in generating a 2D projection that “can reveal meaningful insights about data, e.g., cluster and outliers” [61]. It is more visually interpretable and depending on the distribution, more intuitive than MDS results, which preserve the global structure more at the expense of local structure retained by t -SNE [45]. As shown in Figure 6, each HFL entity in the 2D space represents a pair of “client id/server – round id”. Note that we also consider the server as a special “client” in the 2D space. The servers in the first and last rounds (i.e., server id – first round and server id – last round) are highlighted and HFL servers of all communication rounds are linked by a black curve. Users can click/lasso any entities on the projection space for interaction. We render a contour map in the background to reveal areas with different densities. Intuitively, a normal point tends to lie in high-density areas where many other points have similar behaviors. We use the kernel density estimation (KDE) to define the density.

The summary view (Figure 3(B)) provides statistical information in terms of the number of HFL clients, the number of communication rounds, the growth of the client count and the sampling count, the loss reduction, and the accuracy growth from the initial state to the final one.

6.3 Pairwise Comparison View for Metric Analysis

Although the above projection view conveys additional information about clusters or outliers of participating clients in all communication rounds based on the pair of “client id/server – round id”, it remains unclear how much of it is the noise that is captured in the process as opposed to relevant semantic structure [45], making interpreting and in particular comparison across multiple HFL communication rounds challenging. Furthermore, HFL experts often need to explore the properties of each HFL client in a certain communication round. For example, when assessing the contribution of clients for each communication round, the experts wanted to know their information such as the local model quality, the accuracy, and loss, as well as compare the local model performance among clients. Thus, comparing metrics among HFL clients can help the experts assess the model quality in one or several HFL communication round(s). Note that in this view, we use the same metrics as in the projection view.

Particularly, for a uniform comparison, we set the metrics of the HFL server in the final communication round as the benchmark metrics and compare the metrics of all clients and HFL servers in other communication rounds with the benchmark to obtain the normalized similarities. Thus, we can use the similarities as the normalized metrics for comparison. Note that among the above metrics, three of them, i.e., $Acc_k^{(r)}$, $Loss_k^{(r)}$, and $N_k^{(r)}$ are values and the other two metrics, i.e., $H_k^{model(r)}$ and $H_k^{gradient(r)}$ are value distributions. For example, for metrics like $Acc_k^{(r)}$, $Loss_k^{(r)}$, and $N_k^{(r)}$, we just normalize them by using the same metrics adopted from the HFL server in the final communication round; for the other two metrics $H_k^{model(r)}$ and $H_k^{gradient(r)}$, we calculate the value of RMSE between two histograms to obtain their similarity as the normalized

Fig. 7. (1) Users can sort the rankings of the clients based on the normalized metrics of “loss”, “accuracy”, sample count”, “model similarity”, “gradient similarity”, and “all”, i.e., if we click on “all”, the rankings are based on either Cosine or Euclidean distance by considering all the mentioned metrics. (2) The top row always represents the metrics of the HFL server in different communication rounds. (3) The columns represent the rankings of all the involved clients in the corresponding communication round. The length of the color bars indicates the corresponding normalized value of the metric. (4) The curve area visualizes the similarity change between two adjacent HFL clients.

one. Note that it should be more accurate to use KL-divergence to calculate the distance between two distributions. However, in our case, we transfer the distribution to histograms, i.e., we divide the original distributions into bins for generating the histograms, so we use distance-based metric RMSE to calculate the error. In this way, all the original metrics of clients and servers in different communication rounds are normalized.

Design Consideration. Manually selecting clients for metric comparison is cumbersome, especially when there are many clients involved in the HFL process and no clear visual cues are available for domain experts to decide which clients should be selected for comparison. Therefore, we convert the pairwise comparison to ranking comparison to help assess the properties of clients. To achieve this target, the design should meet the following criteria: (1) *Encode ranking similarity.* The rankings that are determined by pairwise client similarity should be grasped rapidly. (2) *Encode ranking causes.* Users must be able to evaluate the details of each client’s properties to understand how the rankings are determined. (3) *Compare rankings between communication rounds.* Multiple rankings of different communication rounds should be placed into context with one another, thereby allowing domain experts to compare rankings among communication rounds simultaneously.

Visual Designs and Interactions. Inspired by Lineup [62], we design a pairwise comparison view (Figure 7)(R.2, R.4 – R.5) to help domain experts conduct metric comparison at the level of clients. Particularly, we present the metrics of each HFL client as a combined bar (Figure 7(3)), in which the length of a single bar with different colors indicates the normalized metric value of the corresponding client. The top row (Figure 7(2)) in each communication round always represents the HFL server in that communication round. We line up rankings of 5 consecutive communication rounds horizontally. The rankings are based on the value of a certain metric or *Euclidean/Cosine* distance when considering all metrics as a whole (Figure 7(1)). We use *Euclidean/Cosine* distance instead of the previous Canberra Distance to calculate the similarity here due to that all the metric values have been normalized to an equal interval. The similarity is encoded by a curve area that lies on the right side of the rankings to visualize the similarity change between two adjacent clients (Figure 7(4)), i.e., a peak indicates a considerable change of similarity between two adjacent clients, which can resolve the ranking ambiguity of two clients with similar similarity. When clicking on a single client, all identical ones across communication rounds would be highlighted.

6.4 Ranking View for Contribution Analysis

To understand the overall ranking of clients in terms of the previously mentioned metrics, we design a ranking view for contribution analysis. As shown in Figure 3(E)(R.5), the x -axis represents the ranking ids and the y -axis displays the ids of the HFL clients. We use box plots to visualize the ranking distribution for an individual client. A box plot represents the minimal ranking, 25% quantile, median, 75% quantile, and the maximum ranking of a certain HFL client. We obtain the rankings of a certain HFL client by collecting its corresponding rankings in the pairwise comparison view. For example, if we select “Sample Count” in the pairwise comparison view, we obtain the rankings of each HFL client from all its participated communication rounds based on “Sample Count” and plot the box plot distribution for that client. Furthermore, different ranking options, including minimal ranking position (“Min”), maximum ranking position (“Max”), median ranking position (“Median”), and ranking by participating frequency (“Round”) are also provided for sorting.

6.5 Interactions Among the Views

To explore the HFL process, the experts can first load the corresponding HFL logs, and the summary view shows the general information of the HFL process. The projection views provide the experts an intuitive overview of the performance evolution of all the HFL clients and servers in different communication rounds. The experts can click/lasso any interesting nodes, and the HFL overview would jump to the corresponding position to provide visual cues for further inspection. By inspecting the HFL overview, users can identify the communication rounds with potential anomalies by observing relevant metrics, such as the participation status of each HFL client and the accuracy and loss distribution. By clicking on the abnormal communication round, users can witness the detailed metric distribution of each HFL client and server from the pairwise comparison view. The judgment of the clients’ contribution to the performance of the aggregated model can be assessed via the contribution ranking view.

Apart from the defining capabilities of *HFLens*, rich interactions are integrated to catalyze an efficient in-depth analysis. (1) *Filtering and Highlighting.* Users can interactively select one/several client(s) of interest in the projection view for the system to automatically highlight the corresponding information in other views. (2) *Linking and Clicking.* Views are automatically linked through client correspondence. The coordinated interactions among these clients facilitate the examination of various information at different

granularities, such as the HFL overview, communication round, and client instance levels, thus conducting convenient exploration. For example, when clicking on one communication round in the HFL overview, five consecutive rounds before and after the selected round would be displayed in the pairwise comparison view. Users can click on any client in the pairwise comparison view or the node in the projection view for the system to highlight identical clients across the neighboring rankings with highlighted rectangles as shown in Figure 7.

7 EVALUATION

7.1 Case One: 10-class Image Classification by HFL

We first report how the experts familiarized themselves with *HFLens*. We used a well-known dataset and simulated an experiment, which was artificially designed for running a typical HFL process. Specifically, we evaluated our system with a 10-class image classification task over the Fashion-MINIST dataset [63], which contains 60,000 samples in the training data set and 10,000 samples in the testing data set.

Experimental Setting. We simulated 30 HFL clients by using 30 terminals and ran a total of 50 HFL communication rounds. Each client randomly samples its local dataset from the entire training data set and run an independent process as follows.

- 1) The clients can choose when to join or leave the HFL process.
- 2) After receiving the aggregated model from the HFL server, each client randomly chooses a time to start the local fine-tuning process.
- 3) The clients randomly sample data from the training data set and add them to their local training data set.
- 4) In this experiment, we consider two kinds of client anomalies, namely, label-shuffle (i.e., shuffle the labels of training data randomly) and adversarial-model-poison [20] (i.e., malicious clients trying to attain the objective of the targeted misclassification via poisoning the local model parameters). Specifically, in the 10th, 11th, and 12th communication rounds, we randomly choose 10% of the clients to conduct the label-shuffle operation, and in the 20th, 21st, and 22nd communication rounds, we also randomly choose 10% of the clients to conduct the adversarial-model-poison operation.
- 5) Each client performs the local model fine-tuning process for 5 epochs. We use a similar architecture as LeNet-5 [64] to train the image classification model in our HFL process, which is a CNN model with two convolutional layers, both with 3×3 kernels, 32 channels, and 2×2 max-pooling followed by a fully connected layer with 1024 units and a final 10-dimensional *SoftMax* output layer. We use *ReLU* as the activation function in the hidden layers. We then collect the data generated in the running process as described in Table 1 for the subsequent visual analysis.

Overview of HFL Process. After loading the log data organized in Table 1 into *HFLens*, the summary view provides the general information, including the number of HFL clients, the number of communication rounds, the growth of the client and sample counts, the loss reduction, and the accuracy growth (Figure 3(B))(R.1). The complete HFL process of all the involved clients is visualized in Figure 3(C). The experts first identified that the clients participating in the HFL process vary with the evolution of the HFL process. Particularly, the OCR quantifies the network change of the federated process. From the curve, the experts observed several rounds where the federated network

changes significantly. For instance, the network changes remarkably from the 6th to the 7th communication rounds as many clients are joining or leaving the federated network (the colors of the clients are denoted by blue or red, representing the status of either joining or leaving the federated network). They also observed that the sample counts among the clients in each HFL communication round are generally distributed evenly, and the total sampling count continuously increases (R.2). The running duration part displays the starting and ending time of the local fine-tuning process of each HFL client, where they noticed that the starting timestamps of local model fine-tuning of each client vary, which may be caused by potential network delay. The fine-tuning duration indicated by the slope of the lines also varies among the clients, which is related to the local sampling count, that is, a large local sampling count requires a long training time. From this view, the experts decided to adjust the local waiting time for each HFL communication round to better suit the running duration status of the clients for the subsequent communication rounds of the HFL system. From the distributions of loss and accuracy, the experts further observed the loss and accuracy distributions and generally identified that the value of loss continuously decreases and the accuracy continuously increases, except that in the 10th – 12th communication rounds where the accuracy drops remarkably.

Anomaly Detection. To identify the problems in the 10th – 12th communication rounds, the experts clicked the rectangle that represents the 10th communication round as shown in Figure 3(1) and sorted the clients based on the Cosine distance by choosing “All,” which considers the metrics of loss, accuracy, sample count, model similarity and gradient similarity simultaneously (Figure 3(2)). They observed that in the pairwise comparison view (Figure 3(3)), the first three HFL clients in the 10th communication round are considerably different from the others in terms of loss and accuracy, and different from others considering the model and the model gradient (see a break of the *Cosine* similarity curve (Figure 3(4))), implying that these updated models go a wrong direction of model performance. They then clicked a potential anomalous client in the pairwise comparison view (Figure 3(3)), and the corresponding performance curve is shown in Figure 3(5), from which the experts could compare the performance with the overall distribution (R.4). The experts then moved to the projection view and identified that the abnormal client is located in one place (Figure 3(6)), and the other normal ones are clustered in another area (Figure 3(7)) (R.3). All these findings are consistent with our simulation experiment setting in which we deliberately conducted *label-shuffle* in the 10th communication round. The experts also identified a similar issue in the 20th – 22nd communication rounds (Figure 8), which is caused by the *adversarial-model-poison* in the 20th – 22nd communication rounds, thus demonstrating the efficacy of *HFLens*.

Contribution Assessment. The contribution ranking view leverages the design of the box plot to demonstrate the participation status of all the clients in descending order (R.5). Given that the FL local data on clients are totally invisible to the server, the experts, especially E.4, obtained a general idea of the clients’ contribution through the rankings by different factors. Among them, loss and accuracy may possibly reflect the data quality on each client, and the training number represents the data contribution. For example, E.4 wanted to know what contributes to the ranking of the abnormal client observed in Figure 3(E). By ranking the clients using different metrics, they observed that the abnormal clients are not always at the bottom. E.4 commented that “*this is*

Fig. 8. The experts first (1) clicked on the rectangle that represents the 20th communication round which shows a decreasing accuracy of model performance. They then (2) clicked the first HFL client in the 20th communication round to observe its accuracy and loss (3), which are remarkably worse than the other clients, and it locates far away from the normal clients (4).

reasonable since the abnormalities of the clients only last a few rounds.” In most of the other rounds, they behave normally and provide decent performances.

7.2 Case Two: Visual Object Detection by HFL

The second case originates from the HFL-based object detection task adopted from a real-world scenario by our collaboration experts. Particularly, we collected log data from an online HFL system named *FedVision* [50] to inspect its running process. *FedVision* is a HFL-based video detection system that includes surveillance systems from several worksites. HFL is adopted to jointly train an object detection model due to the privacy issue because these worksites belong to different companies. At present, 15 worksites participate in the HFL system. The number of cameras installed varies for different worksites, and the workers periodically (i.e., every 7 days) select some frames from the surveillance video for labeling. Particularly, they would label dangerous objects, such as fire sources, irregular stacking, and sharp objects. The object detection model is based on *YoloV3* [65], which is one of the most used object detection algorithms. *YoloV3* contains 107 layers with 75 convolutional layers. The workers can annotate a given image by specifying bounding boxes and adding corresponding label information in *FedVision*. *FedVision* adopts *Darknet* [66] for annotation. The training process in this case has gone through 40 communication rounds. The total number of clients that participated in the federated learning process is 15. This case demonstrates how the experts identified abnormal issues and then decided to take appropriate interventions.

Identify Abnormal Issues. After loading the log data into *HFLens*, the experts quickly spotted several issues. They first observed that from the first to the final communication rounds of the HFL process, the performance evolution of all HFL clients and servers (i.e., “client id/server – round id”) follows a path, while some points are relatively far away from the path (R.1). The experts maintained that these points are most likely anomalies, so they clicked a point (Figure 9(1)) that highlighted HFL client 1, and the details of its running process are shown in Figure 9(2) (R.3 – R.4). The experts observed an obvious fluctuation in the metric of accuracy from the 18th to the 22nd communication rounds of client 1. By exploring the pairwise comparison view (Figure 9(3)), they found that the accuracy drops frequently, and the loss increases significantly (R.2), which they doubted to be due to the data or

labeling quality issues. After contacting with the corresponding construction worksite, the experts confirmed that the performance fluctuation was caused by the improper operation of a newcomer. With more exploration, the experts also found that the accuracy of client 5 is considerably lower than the median value, as shown in Figure 9(4). Through communication with the corresponding worksite, we found that the deployment scenario of client 5 is quite different from those of the other HFL clients. Particularly, client 5 is a factory surveillance system, whereas the other clients are all construction worksites. On the basis of this finding, the experts suggested to temporarily reduce the aggregation weight of client 5 in the subsequent HFL process and seek more similar scenarios to join the *FedVision* system. Furthermore, the experts observed that OCR exhibits a considerable fluctuation from the 21st. Through communication with the corresponding worksite, we found that the deployment scenario of client 5 is quite different from those of the other HFL clients. Particularly, client 5 is a factory surveillance system, whereas the other clients are all construction worksites. On the basis of this finding, the experts suggested to temporarily reduce the aggregation weight of client 5 in the subsequent HFL process and seek more similar scenarios to join the *FedVision* system. Furthermore, the experts observed that OCR exhibits a considerable fluctuation from the 21st.

Take Interventions. According to the observed issues, the experts decided to make the following adjustment. First, the experts notified the relevant worksites and disseminated the proper labeling method to all the involved colleagues. Second, the experts adjusted the subsequent aggregation strategy in *FedVision* by lowering the aggregation weight of some HFL clients, such as client 5. Particularly, they informed the corresponding worksite of client 5 that its current object detection performance is remarkably lower than that of the other clients and recommended heightened surveillance. Third, the experts emphasized from the OCR curve that the HFL network stability of *FedVision* must be considered because the disconnection of HFL clients might affect the overall model performance.

8 DISCUSSION AND LIMITATION

We conducted a semi-structured interview with the experts (E.1 – E.5) to evaluate *HFLens* and to check whether our approach helps them understand and analyze the HFL process.

System Performance. All experts appreciated the ability of our system to support the interactive exploration of the HFL process. E.1 suggested that our system would greatly boost their work efficiency. Conventionally, they manually extract logs and summarize model performance-related metrics and then check the HFL process. E.4 described his experience of comparing different HFL clients from different communication rounds as trial-and-error. “All I got each round was a single metric and could not dig deeper to find the underlying causes. This is quite stressful and time-consuming.” With our system, they could easily and intuitively compare interesting clients and their metric distributions in different communication rounds. “I can explore different but interconnected information about a single client in various views,” said E.4. They commented that “*HFLens* is very useful because it provides a novel and highly interactive way to uncover the running process in a typical HFL scenario.”

Feedback and Takeaway. E.2 – E.3 commented that “since all the information comes from insensitive data during a typical HFL process, the system can be applied to current mainstream HFL applications.” *HFLens* can also guide the operation and regulation of HFL applications because “it can identify and diagnose potential risks”, said E.4. *HFLens* projects clients at different communication rounds onto a 2D embedding space, which allows the project manager (E.1) to have a general understanding of the distribution and similarity of clients and the performance evolution progress of the aggregated federated model.

Fig. 9. (1) The evolution of HFL clients follows a path, while some points (i.e., client id/server – round id) that are far away from the path are most likely to be anomalies. (2) The experts clicked on the 21st communication round and inspected the metric distributions of each HFL client in the pairwise comparison view. (3) They then identified an abnormal client, i.e., client 1 of which the accuracy is quite low. (4) They also found that the performance of client 5 is always lower than the median value of accuracy.

Visual Design and Interaction. The experts were satisfied with the customized visual design and the interaction capability. We drew inspiration for our design from the observations of these experts’ conventional practices and their responses in the interviews and deliberately selected familiar visual metaphors (e.g., box plots, lines, and bars) to help the experts quickly familiarize themselves with our visual encodings and designs. After introducing them to the basic views and functions of *HFLens*, the experts developed a customized path using this system for exploration purposes.

Generalizability and Scalability. When discussing with the experts about which components of *HFLens* can be directly deployed for other HFL applications and which need customization to explore the potentials of our system, two insights were obtained: (1) *Visual Design*. The experts commented that our designs are quite generic and can fit other application scenarios without much modification. (2) *Data*. *HFLens* covers most of the insensitive data that can be collected by the HFL server. However, with the advancement in HFL research, more data can be derived to understand and control the HFL process, which could be considered for inclusion in *HFLens*. Furthermore, *HFLens* has well-defined APIs and can be easily attached to any existing HFL applications. In terms of scalability, our system currently handles no more than 50 clients, and we are fully aware that if the number of clients further increases, our visualization will have a scalability issue. In this case, we can cluster the clients first before rendering them in the HFL overview.

Limitation. First, we only worked with a small team of experts (E.1 – E.5) for the evaluation and thus could not provide a quantitative assessment of the system. Second, with the advancement in adversarial attacking research, new attacking algorithms, which do not have differentiable performance or histogram, may not be detected by our system. Third, the current contribution ranking view is based on several metrics. More metrics can be included according to different scenarios. Fourth, hardware utilization and clients’ configuration, which do not include private information, are also important factors to be considered in future work, for which

a good visualization may help HFL experts better allocate the computational resources consumed in the local fine-tuning process.

9 CONCLUSION AND FUTURE WORK

In this paper, we introduce the exploratory visual analytics system *HFLens*, which supports the comparative visual interpretation of the HFL process at the overview, communication round, and client-instance levels. Particularly, the system assists domain experts in inspecting the overall running process of HFL servers and clients, correlating clients’ information in multiple communication rounds, identifying potential anomalies, and assessing the contributions of HFL clients. Two case studies and domain experts’ feedback confirm the efficacy of our system. The system can be improved in several ways. First, we plan to perform a systematic evaluation involving additional experts and baseline systems, like *FATEBoard*. Second, we will attempt to include general parameters in machine learning models that can be used in the HFL process for detailed discussion. Third, our system currently aims to analyze the running process for passing communication rounds. If users identify any issues in using our visual analytics system, they could make decisions and take actions to intervene in the corresponding federated learning system. Then, the new logs would be fed into our visual analytics system again to monitor improvement or its absence. For better user experiences, our system must be deployed in real time. We will also include more HFL systems in different application scenarios for field studies.

ACKNOWLEDGMENTS

We are grateful for the valuable feedback and comments provided by Prof. *Qiang Yang* and the anonymous reviewers. This work is partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China under Grant No. C6030-18G.

REFERENCES

- [1] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, 2017. [Online]. Available: <https://arxiv.org/pdf/1602.05629.pdf>
- [2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.
- [3] J. Chen, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting Distributed Synchronous SGD," in *Workshop track-ICLR 2016*, no. 2012, 2016, pp. 1–5. [Online]. Available: <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/45187.pdf>
- [4] A. T. Sebastian Caldas, Jakub Konečný, Brendan McMahan, "Expanding the Reach of Federated Learning by Reducing Client Resource Requirements," in *ICLR 2019 Conference Blind Submission*, 2018, pp. 1–10. [Online]. Available: <https://openreview.net/forum?id=SJlpM3RqKQ>
- [5] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A Benchmark for Federated Settings," no. 1, pp. 1–6, 2018. [Online]. Available: <http://arxiv.org/abs/1812.01097>
- [6] Fedai, "Computer vision Platform powered by Federated Learning," 2019. [Online]. Available: <https://www.fedai.org/cases/computer-vision-platform-powered-by-federated-learning/>
- [7] H. H. Zhuo, W. Feng, Q. Xu, Q. Yang, and Y. Lin, "Federated Reinforcement Learning," *Tech. Rep.*, 2019. [Online]. Available: <http://arxiv.org/abs/1901.08277>
- [8] J. Lee, J. Sun, F. Wang, S. Wang, C. H. Jun, and X. Jiang, "Privacy-preserving patient similarity learning in a federated environment: Development and analysis," *Journal of Medical Internet Research*, vol. 20, no. 4, pp. 1–27, 2018.
- [9] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, "LoAdaBoost: Loss-Based AdaBoost Federated Machine Learning on medical Data," 2018. [Online]. Available: <http://arxiv.org/abs/1811.12629>
- [10] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, and Q. Yang, "HHHFL: Hierarchical Heterogeneous Horizontal Federated Learning for Electroencephalography," pp. 1–7, 2019. [Online]. Available: <http://arxiv.org/abs/1909.05784>
- [11] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of Biomedical Informatics*, vol. 99, 2019. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1903/1903.09296.pdf>
- [12] W. Li, F. Milletari, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. J. Cardoso, and A. Feng, "Privacy-Preserving Federated Brain Tumour Segmentation," pp. 133–141, 2019. [Online]. Available: <http://arxiv.org/abs/1910.00962>
- [13] Mike, "Federated learning: distributed machine learning with data locality and privacy," 2018. [Online]. Available: <https://blog.fastforwardlabs.com/2018/11/14/federated-learning.html>
- [14] T. Fan, "FATE-Board_FATE's Visualization Toolkit," pp. 1–11, 2018. [Online]. Available: <https://github.com/FederatedAI/FATE-Board>
- [15] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," *Tech. Rep.*, 2016. [Online]. Available: <http://arxiv.org/abs/1610.05492>
- [16] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated Optimization: Distributed Machine Learning for On-Device Intelligence," 2016. [Online]. Available: <http://arxiv.org/abs/1610.02527>
- [17] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," *Annual Reviews of Psychology*, vol. 51, no. 2006, pp. 665–697, 2017. [Online]. Available: <https://arxiv.org/abs/1712.07557>
- [18] M. Andrychowicz, G. Deepmind, and K. Kurach, "Efficient Federated Learning Via Variational Dropout," pp. 1–12, 2019.
- [19] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal Client Behavior Detection in Federated Learning," vol. 2019, no. i, 2019. [Online]. Available: <http://arxiv.org/abs/1910.09933>
- [20] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing Federated Learning through an Adversarial Lens," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97. PMLR, 2019, pp. 634–643. [Online]. Available: <http://arxiv.org/abs/1811.12470>
- [21] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated Learning for Mobile Keyboard Prediction," 2018. [Online]. Available: <http://arxiv.org/abs/1811.03604>
- [22] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards Federated Learning at Scale: System Design," *Tech. Rep.*, 2019. [Online]. Available: <http://arxiv.org/abs/1902.01046>
- [23] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017. [Online]. Available: <http://arxiv.org/abs/1711.10677>
- [24] R. Nock, S. Hardy, W. Henecka, H. Ivey-Law, G. Patrini, G. Smith, and B. Thorne, "Entity Resolution and Federated Learning get a Federated Resolution," 2018. [Online]. Available: <http://arxiv.org/abs/1803.04035>
- [25] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [26] Y. Liu, T. Chen, and Q. Yang, "Secure Federated Transfer Learning," *Tech. Rep.*, 2018. [Online]. Available: <http://arxiv.org/abs/1812.03337>
- [27] M. Rojek, "Devices learning from each other ? See it live this September at AI Summit in San Francisco !" pp. 7–11, 2018.
- [28] X. Wei, Q. Li, Y. Liu, H. Yu, T. Chen, and Q. Yang, "Multi-Agent Visualization for Explaining Federated Learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, {IJCAI-19}*. International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 6572–6574.
- [29] Y. MING, "A survey on visualization for explainable classifiers," 2017.
- [30] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.
- [31] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
- [32] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.
- [33] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in neural information processing systems*, 2017, pp. 4765–4774.
- [34] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015.
- [35] G. Wang, "Interpret federated learning with shapley values," *arXiv preprint arXiv:1905.04519*, 2019.
- [36] C. Molnar, *Interpretable machine learning*. Lulu. com, 2019.
- [37] J. D. Olden and D. A. Jackson, "Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks," *Ecological modelling*, vol. 154, no. 1-2, pp. 135–150, 2002.
- [38] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding projector: Interactive visualization and interpretation of embeddings," *arXiv preprint arXiv:1611.05469*, 2016.
- [39] F. Hohman, K. Wongsuphasawat, M. B. Kery, and K. Patel, "Understanding and visualizing data iteration in machine learning," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2020.
- [40] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg, "Visualizing dataflow graphs of deep learning models in tensorflow," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 1–12, 2017.
- [41] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 91–100, 2016.
- [42] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Do convolutional neural networks learn class hierarchy?" *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 152–162, 2017.
- [43] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson, "The what-if tool: Interactive probing of machine learning models," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 56–65, 2019.
- [44] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting interactive performance analysis for multiclass classifiers," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 61–70, 2016.
- [45] Q. Li, K. S. Njotoprawiro, H. Haleem, Q. Chen, C. Yi, and X. Ma, "Embeddingvis: A visual analytics approach to comparative network embedding inspection," in *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2018, pp. 48–59.
- [46] M. T. Ribeiro, S. Singh, and C. Guestrin, "why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

- [47] C. Ilias and S. Georgios, "Machine learning for all: A more robust federated learning framework," *ICISSP 2019 - Proceedings of the 5th International Conference on Information Systems Security and Privacy*, pp. 544–551, 2019.
- [48] G. Wang, C. X. Dang, and Z. Zhou, "Measure Contribution of Participants in Federated Learning," 2019. [Online]. Available: <http://arxiv.org/abs/1909.08525>
- [49] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," *Proceedings - 2019 IEEE VTS Asia Pacific Wireless Communications Symposium, APWCS 2019*, pp. 1–5, 2019.
- [50] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "Fedvision: An online visual object detection platform powered by federated learning," *arXiv preprint arXiv:2001.06202*, 2020.
- [51] Pascal Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1592, pp. 123–139, 1999.
- [52] L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients," *NeurIPS 2019*, 2019. [Online]. Available: <http://arxiv.org/abs/1906.08935>
- [53] P. McLachlan, T. Munzner, E. Koutsofios, and S. North, "Liverac: interactive visual exploration of system management time-series data," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 1483–1492.
- [54] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 1996, pp. 336–343.
- [55] R. Munroe, "Movie narrative charts," *Diagram* available at <https://xkcd.com/657>, 2009.
- [56] P. Federico, J. Pfeffer, W. Aigner, S. Miksch, and L. Zenk, "Visual analysis of dynamic networks using change centrality," *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, pp. 179–183, 2012.
- [57] F. Heimerl and M. Gleicher, "Interactive analysis of word vector embeddings," in *Computer Graphics Forum*, vol. 37, no. 3. Wiley Online Library, 2018, pp. 253–265.
- [58] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea, "Visualizing the hidden activity of artificial neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 101–110, 2016.
- [59] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Netsimile: A scalable approach to size-independent network similarity," *Computer Science*, vol. 12, no. 1, pp. 28(1–28), 2012.
- [60] Q. Li, Q. Shen, Y. Ming, P. Xu, Y. Wang, X. Ma, and H. Qu, "A visual analytics approach for understanding egocentric intimacy network evolution and impact propagation in mmorpgs," in *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2017, pp. 31–40.
- [61] M. Kim, M. Choi, S. Lee, J. Tang, H. Park, and J. Choo, "Pixelsne: Visualizing fast with just enough precision via pixel-aligned stochastic neighbor embedding," *arXiv preprint arXiv:1611.02568*, 2016.
- [62] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "Lineup: Visual analysis of multi-attribute rankings," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2277–2286, 2013.
- [63] H. Xiao, K. Rasul, and R. Vollgraf, (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
- [64] Y. LeCun et al., "Lenet-5, convolutional neural networks," URL: <http://yann.lecun.com/exdb/lenet>, vol. 20, p. 5, 2015.
- [65] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [66] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

Quan Li is an assistant professor in School of Information Science and Technology at ShanghaiTech University. He received the PhD degree from the Hong Kong University of Science and Technology. He was a research scientist at AI Group, WeBank, Shenzhen. He obtained his Bachelor's Degree from Wuhan University and Master's Degree from Tsinghua University. For more details, please refer to <https://thinkavatar.github.io/>.

Xiguang Wei received his PhD degree from the University of Hong Kong. He is now a research scientist at Shenzhen Semacare Medical Technology Co., Ltd. His research interest includes federated learning, bioinformatics, medical image processing. He obtained the Bachelor's Degree from China Agricultural University and Master's Degree from the Chinese University of Hong Kong.

Huanbin Lin graduated from Guangdong University of Technology, majoring in digital media technology. He previously worked in Tencent, and was responsible for QQ platform development. He is now an engineer at AI Group, WeBank, Shenzhen, and participates in both engineering and research in the field of AI interactive visual analytics and visualization.

Yang Liu is a senior researcher of AI department at WeBank. Her main research interests include machine learning, federated learning, transfer learning, and multi-intelligent system, etc. Before joining WeBank, she worked as a data scientist at Dataminr, an American startup, where she was responsible for the research and development of algorithms and products. She obtained her Ph.D. from Princeton University, and graduated from Tsinghua University.

Tianjian Chen is deputy general manager of WeBank AI Group. He is now responsible for building a banking smart ecosystem based on federated learning and responsible for federal ecosystem and open source project FATE. He served as chief architect of Baidu Financial Services and Baidu Chief architect. He has over 12 years of extensive distributed system design experience and has achieved technological innovation in many applications such as web search engine, peer-to-peer network storage, genomics, recommendation system, digital banking and machine learning.

Xiaojuan Ma is an assistant professor of Human-Computer Interaction (HCI) at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. She received her PhD in Computer Science at Princeton University. She was a post-doctoral researcher at the Human-Computer Interaction Institute of Carnegie Mellon University, and before that a research fellow in the National University of Singapore in the Information Systems department. Before joining HKUST, she was a researcher of Human-Computer Interaction at Noah's Ark Lab, Huawei Tech. Investment Co., Ltd. in Hong Kong.