Fast, Robust and Interpretable Participant Contribution Estimation for Federated Learning

Yong Wang[1], Kaiyu Li[1*], Yuyu Luo[2], Guoliang Li[1*], Yunyan Guo[1], Zhuo Wang[1]
[1] *Tsinghua University,* [2]*HKUST (GZ)*
{wangy18@mails., liguoliang@, yunyanguo@mail., wang-z18@mails.}tsinghua.edu.cn,
ky-li18@tsinghua.org.cn, yuyuluo@hkust-gz.edu.cn

***Abstract*—In this paper, we introduce CTFL, a fair, robust, and interpretable framework designed to estimate clients' contributions to federated learning, aiming to incentivize high-quality data providers to participate in the federation. Firstly, CTFL can precisely allocate contribution credits in a single pass of model training and inference, ensuring computational efficiency. This is accomplished by tracking the test performance gain brought by each participant through exploiting classification rules. Secondly, CTFL adheres to essential theoretical properties of an ideal contribution estimation algorithm, including symmetry, zero-element, and additivity, ensuring fair and rational estimations. Thirdly, CTFL demonstrates resilience against strategic and malicious behaviors due to carefully crafted micro and macro contribution estimation schemes. Fourthly, CTFL offers insights into participants' roles within the federation by interpreting their contribution scores through respective high-frequently activated rules. Finally, CTFL integrates logical neural networks and model binarization techniques to ensure effectiveness and efficiency while preserving data privacy. Extensive experiments validate that CTFL accurately estimates contributions, significantly reducing computation time by 2-3 orders of magnitude compared to state-of-the-art methods while maintaining robustness.**

***Index Terms*—data valuation, federated learning, contribution estimation, interpretable machine learning**

I. Introduction

Federated Learning (FL) is a collaborative machine learning paradigm [1] wherein multiple participants (clients) jointly train a global model while protecting their data privacy, which has garnered significant attention in both the database [2]–[6] and machine learning [1], [7]–[9] communities. In practice, the clients may be reluctant to share their data due to a lack of incentives or feel unfair about the revenue allocation results [10], [11]. Thus, developing fair and transparent contribution estimation mechanisms in FL is crucial to motivating clients to participate and subsequently maintaining the federation's sustainability. This paper studies the problem of assigning scores to all the clients in horizontal FL, indicating the importance of their respective datasets, where clients share a common feature space but possess distinct data instances.

Challenges. In practice, the primary challenges of contribution estimation in horizontal FL arise from three facets.

(C1) High Computational Costs. Most existing approaches necessitate training and evaluating a considerable number of coalitions (combinations of instances from distinct clients), for estimating clients' contributions. For instance, the widely used FL contribution estimation method, `ShapleyValue` [12], requires evaluating 2^n coalitions to compute clients' contributions, where n denotes the number of participants. Efforts to address this challenge involve optimization techniques such as strategic sampling [9] and data distribution assumptions [13], [14], aiming to reduce the number of evaluated coalitions to a polynomial order [15]. Despite these strategies, the computational overhead remains substantial and impractical in comparison to the original FL model training, especially when dealing with a large number of clients.

(C2) Susceptible to Adverse Behaviors. In practice, adverse participants may attempt to gain additional contribution credits by duplicating data, incorporating inaccurately labelled records, or intentionally introducing flipped labels to disrupt the federation. Identifying and defending against such adverse behaviors is essential for precise contribution estimations and the construction of robust FL models However, current methodologies fall short in identifying these malicious behaviors [16], primarily because each participant's data is usually evaluated as a whole using an abstracted utility metric.

(C3) Poor interpretability. Exsiting methods cannot provide evidential explanations for estimation results [17], offering only numerical scores without reasoning. This limitation arises from the inherent opacity of black-box models that conceal the computation of data utility, compounded by the fact that participants do not have access to each other's datasets. Consequently, the federation is unable to convince low-contributing participants, or guide them to enhance their data quality.

Our Methodology. To address the challenges mentioned above, we propose a novel contribution estimation framework named CTFL (Contribution Tracing for Federated Learning), which achieves fast, fair, robust, and interpretable contribution estimation by fully exploiting rule-based models. We compare CTFL to existing methods in Table I, based on the experimental results on benchmarks in Section VI.

First, CTFL achieves fast contribution estimation via just *a single pass of rule-based model training and inference* [18], while ensuring properties of group rationality, symmetry, zero element and additivity. Specifically, we train a global model on the data of all clients and then trace the test performance gain brought by each client based on the activated classifi-

[*] Guoliang Li and Kaiyu Li are the corresponding authors.

TABLE I: Comparing CTFL to Existing Approaches

Method	Accuracy	Efficiency	Robust-ness	Inter-pretable
Individual [20], [21]	+	+++	+++	×
LeaveOneOut [22]	+	++	+	×
LeastCore [23]	++	+	++	×
ShapleyValue [9], [24]	+++	+	+	×
CTFL (ours)	+++	+++	+++	✓

cation rules (addressing **C1**). Furthermore, the computation process of CTFL can be fully parallelized and speeded up by employing frequent rule sets searching algorithms [19].

Second, unlike existing methods, CTFL is robust to the aforementioned adverse behaviors, which is attributed to judiciously designed micro and macro contribution allocation schemes (addressing **C2**). Specifically, our approach does not assign additional contribution credits to those who replicate data, recognize low-quality data by filtering training data that cannot be traced by test instances, and identifies label-flipped data by inspecting the performance loss brought by training data with similar activated rules but contradictory labels.

Third, CTFL enables the interpretation of clients' contribution scores by frequently activated classification rules (addressing **C3**). We summarize the beneficial and harmful characteristics of each client by tracing the corresponding frequently activated rules that lead to correct or miss classifications, respectively. In addition, CTFL can guide active data collection for test cases that are not sufficiently covered by existing training data by summarizing the frequent activation rules of these failed test data. Last but not least, to ensure the generalization and accuracy of the contribution estimation, while preserving data privacy for real-world scenarios, we develop a practical rule-based model integrating logical neural networks [25] and model binarization techniques [26], [27].

Contributions. We make several notable contributions:

(1) *Framework.* We design a novel contribution estimation framework for federated learning called CTFL (Section III-A). CTFL achieves *fast* and *accurate* contribution estimation using rule-based models (Section III-B & III-C). Moreover, CTFL satisfies essential theoretical properties that ensure fair and rational estimations (Section III-D).

(2) *Robustness.* CTFL is robust to adverse behaviors including data replication, low-quality data, and label flipping, owning to its multi-grained estimation schemes (Section IV-A).

(3) *Interpretability.* CTFL interprets participants' contribution scores via their high-frequent rule activations, provides valuable insights into each participant's role, and further guides active data collection for the federation (Section IV-B).

(4) *Implementation.* We develop a practical rule-based model with high generalization ability and precise contribution tracing capability while preserving data privacy (Section V).

(5) *Experiments.* We conduct extensive experiments on four public benchmark datasets. Our results show that CTFL outperforms existing approaches, and remarkably, CTFL accurately estimates clients' contributions with 2-3 orders of magnitude less time than state-of-the-art methods (Section VI).

TABLE II: Model Test Acc. Across Different Participant Sets

Participant Set	$\emptyset$	A	B	C	A, B	A, C	B, C	A, B, C
v: Test Acc. (%)	50	80	80	65	80	90	90	90

II. Contribution Estimation in FL

A. Problem Formalization

Before defining the contribution estimation problem, we first introduce how to quantify data quality using a specific metric.

Definition II.1 (Data Utility Metric)**.** A data utility metric $v : \mathbb{D} \to \mathbb{R}$ is a function that outputs the data utility $v(D)$ of a dataset $D \in \mathbb{D}$ to a specific FL task.

To be rational and accurate, $v(D)$ should be commensurate with the performance improvement that D provides to the task. We focus specifically on classification tasks and adopt the typical model test accuracy to quantify data utility. This approach can be extended to regression tasks and other performance metrics, such as F1-score, or even task-independent metrics, such as data quantity and participant reputation. Let D be a training dataset and $D_{te} = \{(\mathbf{x}_{te}, y_{te})\}$ be a test dataset, which is collected and reserved by the federation. The utility of D can be computed as follows.

$$v(D) = \frac{1}{|D_{te}|} \sum_{(\mathbf{x}_{te}, y_{te}) \in D_{te}} \mathbf{1}[y_{te} = \mathcal{M}_D(\mathbf{x}_{te})] \qquad (1)$$

where $|D_{te}|$ is the test data size, $\mathcal{M}_D(\mathbf{x}_{te})$ is the inference label of $\mathbf{x}_{te}$ by a model trained on D, and $\mathbf{1}[\cdot]$ is an indicator function that outputs 1 if the inference is correct and 0 otherwise. Based on a data utility metric, we can define the contribution estimation problem.

Definition II.2 (Participant Contribution Estimation)**.** Given an FL setting with n participants $N = \{1, \cdots, n\}$, each holding a private local dataset D_i, and a data utility metric v, the participant contribution estimation problem aims to compute a contribution score vector $(\phi_v(1), \cdots, \phi_v(n))$ for the participants. The contribution scores are computed according to a contribution allocation scheme $\phi_v : N \to \mathbb{R}$ that assigns each participant a score based on v.

Example II.1 (Participant Contribution Estimation)**.** Suppose there are three participants A, B and C. A and B hold similar and sufficient typical data, while C holds a small amount of complementary task-critical data. We adopt model test accuracy as the data utility metric and show the test accuracy of models learned on the data of different combinations of A, B and C in Table II. Suppose we use the contribution estimation schemes described in Section II-B. If we use Individual scheme, C's score will be underestimated, *i.e.,* $\phi_v(C) = 0.65$. Alternatively, if LeaveOneOut scheme is adopted, A and B will be considered as zero contributions due to their substitutability. A more reasonable scheme that considers participants' expected marginal test gain is ShapleyValue, which gets $\phi_v(A) = \phi_v(B) = 11.7$, and $\phi_v(C) = 16.6$.

B. Related Work

1) Individual Scheme.: The Individual scheme uses a participant's individual data value as its potential contribution

TABLE III: Table of Notations

Notation	Definition
$N = \{1, \cdots, n\}$	Participants in FL
$D_S, S \subseteq N$	The training data of participants in S
$(\mathbf{x}_{tr}, y_{tr}), (\mathbf{x}_{te}, y_{te})$	A training and a test data instance
$v : \mathbb{D} \rightarrow \mathbb{R}$	A data utility metric
$\phi_v : N \rightarrow \mathbb{R}$	A contribution allocation scheme
$\phi_v(i), i \in N$	Participant i's contribution score
$\mathcal{M}$ ($\mathcal{M}_D$)	A task model (trained on D)
$\mathbf{r}^+, \mathbf{r}^-$	Positive and negative rule vectors
$\mathbf{w}^+, \mathbf{w}^-$	Weight vectors of $\mathbf{r}^+$ and $\mathbf{r}^-$
$ct(\mathbf{x}_{te}, y_{te})$	Training data related to the test case

to FL as stated in [21]. Formally, participant i's contribution $\phi_v(i)$ is computed as $\phi_v(i) = v(D_i), \forall i \in N$. Despite its simplicity and efficiency [28]–[35], the `Individual` scheme is not extensively embraced, due to its limitation in capturing the marginal contribution of participants in FL, therefore can not comprehensively reflect its significance within the FL.

2) Leave-one-out Scheme.: As employed for cross-validation in machine learning tasks [36], `LeaveOneOut` suggests that a participant's contribution is the performance loss incurred by removing it from FL [22]. However, this scheme is unfair to participants with homogeneous data with similar distributions, because removing a single participant from two participants sharing similar data would not result in significant performance loss.

3) Shapley Value Scheme.: `ShapleyValue` [12], [24], [37] quantifies a participant's contribution based on the expected marginal gain derived from incorporating this participant into all the possible combinations of other participants, i.e., $\phi_v(i) = \mathbb{E}_{S \subseteq N \setminus i}[v(D_{S \cup \{i\}}) - v(D_S)]$, $\forall i \in N$, where $\mathbb{E}$ is the expectation function. Although the result of `ShapleyValue` is accurate, it is computationally expensive because of evaluating exponentially many participant combinations, despite many techniques being proposed to optimize the computation process [9], [15].

4) Least Core Scheme.: According to `LeastCore` [38], the contribution sum of any participant subgroup should be at least the subgroup's data value, and the maximum subgroup deficit (*i.e.,* the gap between credits and data values) should be minimized, *i.e.,*,

$$\min e \text{ s.t. } \sum_{i \in S} \phi_v(i) + e \geq v(D_S), \forall S \subseteq N \quad (2)$$

where scores subject to $\sum_{i \in N} \phi_v(i) = v(D_N)$, and e is the optimization objective (or deficit). `LeastCore` is adopted in FL [23] due to its stability from economic perspectives [39], *i.e.,* any potential coalition gets the largest possible credits as optimized. As a consequence, its contribution scores are not fair for individuals as they are not maximized with respect to participants' marginal contributions.

Previous schemes either lack fairness and rationality, or are inefficient in practical scenarios despite acceleration attempts. Moreover, they overlook crucial robustness and interpretability aspects. In contrast, CTFL is fast with only a single pass of model training, and is robust and interpretable by attributing the credits through activated rules.

III. Contribution Estimation with CTFL

A. An Overview of CTFL

We present our system, Contribution Tracing for Federated Learning (CTFL), which achieves fast, accurate, robust, and interpretable contribution estimation, as demonstrated in Figure 1. First, we employ rule-based models to avoid training and evaluating multiple models in computing participants' contributions (step-① in Figure 1). This approach enables us to *train only a single global model on the data of all participants* and then trace each participant's marginal contribution to the model's test performance (Section III-B). Second, we propose a rule-based tracing strategy to match (or identify) the beneficial training data for each correctly classified test instance based on the activated rules (step-② in Figure 1). We then design two contribution allocation schemes to accurately allocate credits to participants according to their beneficial training data (step-③ in Figure 1, Section III-C). Furthermore, CTFL satisfies four theoretical properties that ensure fair and rational contribution estimations (Section III-D). Third, we demonstrate that CTFL is robust against adverse behaviors by fully exploiting the stringent contribution tracing procedure (Section IV-A). Fourth, we interpret the beneficial and harmful characteristics and identify the useless training data for each participant based on their rule activation frequencies (step-④ in Figure 1, Section IV-B). Last but not least, equipping a rule-based model as a task model in CTFL raises concerns regarding model generalizability, rule-based tracing preciseness, and privacy risks. To address these issues, we develop a *practical* rule-based model utilizing logical neural networks and model binarization techniques. This approach achieves high generalization ability, precise tracing capability, and effectively addresses privacy concerns (Section V).

B. Rule-based Models

Most existing approaches require training and evaluating multiple task models to compute the marginal performance gain brought by each participant. However, this method is time-consuming and resource-intensive, making it impractical in real-world scenarios. To address this problem, we propose training a single model on the data of all participants and directly tracing the model test performance (or accuracy) brought by each participant. We achieve this by using rule-based models to monitor the activated rules of the test and training data and trace the test accuracy gain contributed by each participant. The rationale behind this approach is that it allows us to identify the model's inference basis via activated rules, which enables us to pinpoint the training data that contributes to the inference basis.

This idea is inspired by doctors' diagnoses, which rely on summarized symptoms from historical cases. When diagnosing a patient, the doctor determines the disease type by comparing symptoms to those of previous patients. This process is similar to contribution tracing, where the training data from all participants represents the historical cases, the learned model acts as the doctor, the activated rules represent the symptoms, and

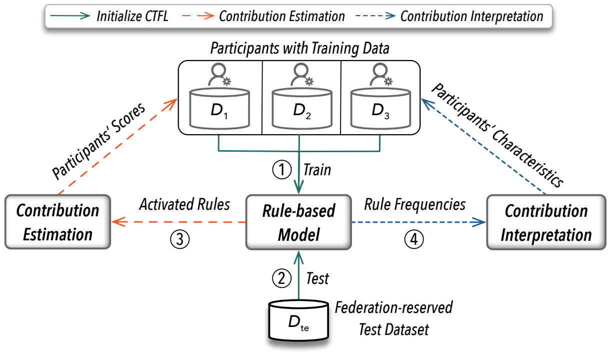


Fig. 1: An Overview of CTFL

the reserved test data for performance evaluation represents new patients. Therefore, we determine the beneficial training data that corresponds to correctly classified test instances using activated classification rules. Based on these intuitions, we present the rule-based models. Meanwhile, this evaluation process is efficient because it only requires training and evaluating a single model.

Definition III.1 (Rule)**.** A rule r, in terms of classification, is a logical formula composed of one or more logical predicates that relate to the values of the input features. For an input $\mathbf{x}$, $r(\mathbf{x}) = 1$ if $\mathbf{x}$ fulfills r's logical predicates, *i.e., r* is *activated*, and $r(\mathbf{x}) = 0$ otherwise.

Each rule explicitly associates with (or supports) an output class label. Without loss of generality, we restrict our discussion to binary classification for ease of presentation, which can be extended to multi-class with minor changes. In this context, r can either be associated with the positive class label or the negative class label, *i.e.,* be positive (denoted as r^+) or negative (denoted as r^-), respectively. We specifically consider atomic predicates $>, <, \leq, \geq$ for continuous features and $=, \neq$ for discrete features, and support three types of operations: conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$).

Example III.1 (Rule)**.** Figure 2 (c) shows four rules generated for an income-level prediction task. For instance, r_1^+ : $capital\text{-}gain > 21k$ is a single-predicate rule that supports the positive label (*i.e.,* high income), and r_2^- : $work\text{-}hours > 14 \vee marital\text{-}status = never$ is a rule of two predicates under disjunction that supports the negative label (*i.e.,* low income). Note that logical operations can be recursively applied to produce compound rules.

In this paper, the rules are derived from the training data automatically using logical neural network [40] by seamlessly blending the learning capabilities of neural networks with the structured reasoning of symbolic logic, enabling learning complex relationships in the data as understandable rules (Section V). We are now ready to describe rule-based task models which classify input data using the weighted sum of a collection of activated positive and negative rules.

Definition III.2 (Rule-based Model)**.** A rule-based model $\mathcal{M}$ is defined as a set of rules that are used for classification. The model consists of a positive rule vector $\mathbf{r}^+$ and a negative rule vector $\mathbf{r}^-$, each of length m^+ and m^-, respectively. The positive and negative rule vectors are associated with importance weights $\mathbf{w}^+$ and $\mathbf{w}^-$, respectively. The classification of an input $\mathbf{x}$ is performed using a weighted voting mechanism:

$$\mathcal{M}(\mathbf{x}) = \mathbf{1}[\mathbf{w}^+ \cdot \mathbf{r}^+(\mathbf{x}) \geq \mathbf{w}^- \cdot \mathbf{r}^-(\mathbf{x})] \qquad (3)$$

Utilizing the set $\mathcal{M}$, an input x is categorized as positive if the weighted sum of activated positive rules surpasses the weighted sum of activated negative rules; otherwise, it is classified as negative (as illustrated in Figure 2 (a)).

Example III.2 (Rule-based Model)**.** Recall the rules in Figure 2 (c). Suppose that the rules' weights are $w_1^+ = w_2^+ = w_1^- = 1, w_2^- = 0.5$, and test instance $(\mathbf{x}_{te}^{(1)}, y_{te}^{(1)} = 1)$ have r_2^+ and r_2^- activated. We have $\hat{y}_{te} = \mathcal{M}(\mathbf{x}_{\mathbf{te}}^{(\mathbf{1})}) = \mathbf{1}[\mathbf{w}_\mathbf{2}^+ * \mathbf{r}_\mathbf{2}^+ \geq \mathbf{w}_\mathbf{2}^- * \mathbf{r}_\mathbf{2}^-] = \mathbf{1}[1 > 0.5] = y_{te}$.

Significantly, the aforementioned rule-based models are recognized as rule-based ensemble classifiers [41], wherein learned biases are typically incorporated before employing the indicator function to accommodate non-linear relationships (details omitted for clarity). To attain robust generalization and accurate contribution tracing capabilities while upholding data privacy, we will introduce a practical implementation of a rule-based model utilizing logical neural networks and model binarization techniques in Section V.

C. Tracing Participants' Contributions

CTFL estimates participants' contributions by training *only a single rule-based model* on the data of all participants and then tracing the model test performance gain brought by each participant using activated rules. To accomplish this, we first establish the tracing principles with respect to class labels. In particular, when a test instance is correctly classified, the training data (a) in the same class and (b) with similar activated rules on this class are related and considered beneficial. On the other hand, when a test instance is misclassified, then the training data (a) in the wrong class and (b) with similar activated rules on this class are related and considered detrimental. Specifically, given a rule-based model $\mathcal{M}_{D_N}$ and a test instance $(\mathbf{x}_{te}, y_{te})$, there are four tracing cases:

1) *Case-1: True Positive (TP).* $\mathcal{M}_{D_N}$ correctly classifies $\mathbf{x}_{te}$ as positive. In this case, participants who have training data that learn the activated *positive* rules which correctly classify $\mathbf{x}_{te}$, receive a positive credit. For instance, in Figure 2-(b), if the positive rules activated by $\mathbf{x}_{te}$ is r_2^+, then participant A, with positive training data that learn r_2^+, is awarded as TP.

2) *Case-2: True Negative (TN).* $\mathcal{M}_{D_N}$ correctly classifies $\mathbf{x}_{te}$ as negative. In this case, participants who have training data that learn the activated *negative* rules which correctly classify $\mathbf{x}_{te}$, receive a positive credit. Similarly, in Figure 2, TN credit awards to participants B and C which learn r_1^- and r_2^-.

3) *Case-3: False Positive (FP).* $\mathcal{M}_{D_N}$ misclassifies $\mathbf{x}_{te}$ as positive. In this case, participants with training data that learn the activated *positive* rules, is responsible for the performance loss and receive a negative credit. For instance, in Figure 2, FP credit corresponds to participants which learn r_m^+ (no one matched in this example).

4) *Case-4: False Negative (FN).* $\mathcal{M}_{D_N}$ misclassifies $\mathbf{x}_{te}$ as negative. In this case, participants with training data that learn

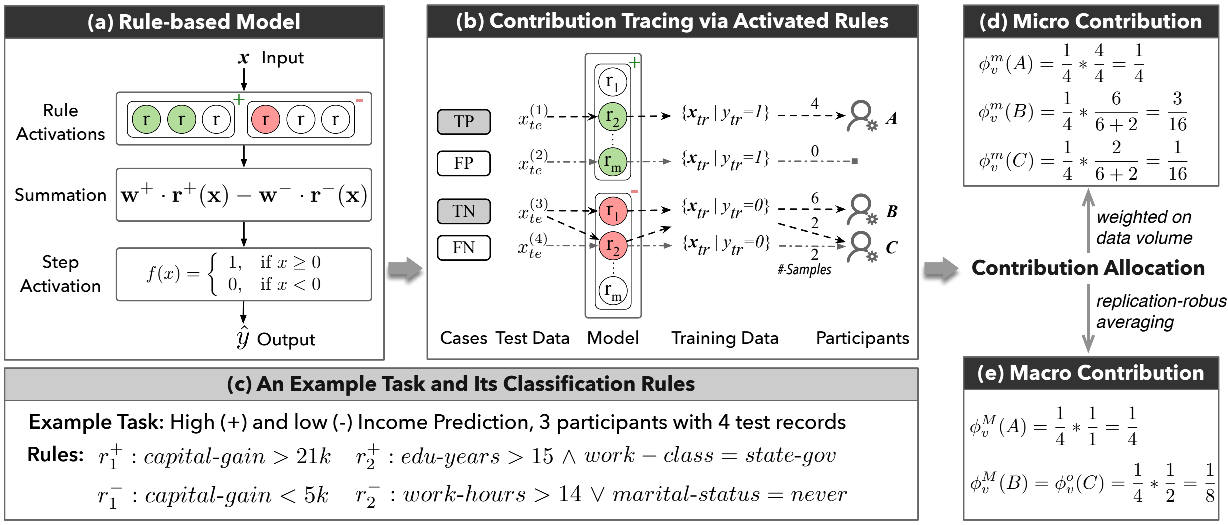

(a) Rule-based Model
x Input
Rule Activations
r r r
r r r
+
-
Summation
$\mathbf{w}^+ \cdot \mathbf{r}^+(\mathbf{x}) - \mathbf{w}^- \cdot \mathbf{r}^-(\mathbf{x})$
Step Activation
$f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$
$\hat{y}$ Output
(b) Contribution Tracing via Activated Rules
TP
FP
TN
FN
$x_{te}^{(1)}$
$x_{te}^{(2)}$
$x_{te}^{(3)}$
$x_{te}^{(4)}$
r_1
r_2
r_m
r_1
r_2
r_m
$\{x_{tr} \mid y_{tr}=1\}$
$\{x_{tr} \mid y_{tr}=1\}$
$\{x_{tr} \mid y_{tr}=0\}$
$\{x_{tr} \mid y_{tr}=0\}$
4
0
6
2
2
#-Samples
A
B
C
Cases
Test Data
Model
Training Data
Participants
(c) An Example Task and Its Classification Rules
Example Task: High (+) and low (-) Income Prediction, 3 participants with 4 test records
Rules: r_1^+ : *capital-gain* $> 21k$ r_2^+ : *edu-years* $> 15 \wedge$ *work* $-$ *class* $=$ *state-gov*
r_1^- : *capital-gain* $< 5k$ r_2^- : *work-hours* $> 14 \vee$ *marital-status* $=$ *never*
(d) Micro Contribution
$\phi_v^m(A) = \frac{1}{4} * \frac{4}{4} = \frac{1}{4}$
$\phi_v^m(B) = \frac{1}{4} * \frac{6}{6+2} = \frac{3}{16}$
$\phi_v^m(C) = \frac{1}{4} * \frac{2}{6+2} = \frac{1}{16}$
weighted on data volume
Contribution Allocation
replication-robust averaging
(e) Macro Contribution
$\phi_v^M(A) = \frac{1}{4} * \frac{1}{1} = \frac{1}{4}$
$\phi_v^M(B) = \phi_v^o(C) = \frac{1}{4} * \frac{1}{2} = \frac{1}{8}$

Fig. 2: Tracing Participants' Contributions Using Rule-based Models

the activated *negative* rules, is responsible for the performance loss and receive a negative credit. Similarly, in Figure 2, FN credit corresponds to participant C which learns r_2^-.

Based on the above contribution tracing principles, we focus on tracing cases TP and TN with correct classifications for contribution computation as we use test accuracy as the data utility metric. Additionally, we consider cases FP and FN to identify malicious participants, which will be described in Section IV-A.

Rule-based Tracing. According to the above tracing principles, we now compute the related training data $ct(\mathbf{x}_{te}, y_{te})$ for each test instance $(\mathbf{x}_{te}, y_{te}) \in D_{te}$. Intuitively, in a strict tracing manner, a training instance (x_{tr}, y_{tr}) is related to (x_{te}, y_{te}) if (a) $y_{tr} = y_{te}$ and (b) the activated rules of x_{tr} include all the activated rules of x_{te} that support y_{te}. However, this strategy is too strict and overlooks the training data that only contribute to a core subset of the corresponding rules. Therefore, we present a softer rule-based tracing strategy to lower the barrier and overcome this limitation for complex scenarios. Specifically, a training instance (x_{tr}, y_{tr}) is related to $\mathbf{x}_{te}$ if (a) $y_{tr} = y_{te}$ and (b) the activated rules of x_{tr} comprise more than τ_w of the activated rules of $\mathbf{x}_{te}$ that support y_{te} (e.g., $\tau_w = 0.6$), where τ_w is a predefined threshold:

$$ct(\mathbf{x}_{te}, y_{te}, \tau_w) = \{(\mathbf{x}_{tr}, y_{tr}) \in D_N | y_{tr} = y_{te}, \text{ and } \frac{\mathbf{w}^* \odot \mathbf{r}^*(\mathbf{x}_{tr}) \cdot \mathbf{r}^*(\mathbf{x}_{te})}{\mathbf{w}^* \cdot \mathbf{r}^*(\mathbf{x}_{te})} \geq \tau_w\} \tag{4}$$

where $D_N = \bigcup_{i \in N} D_i$, r^* (w^*) refers to r^+ (w^+) if $y_{te} = 1$ and r^- (w^-) otherwise, $\mathbf{w}^* \cdot \mathbf{r}^*(\mathbf{x}_{te})$ is the number of activated rules of x_{te}, $\odot$ denotes element-wise multiplication and $\mathbf{w}^* \odot \mathbf{r}^*(\mathbf{x}_{tr}) \cdot \mathbf{r}^*(\mathbf{x}_{te})$ is the number of Intersecting activated rules between $\mathbf{x}_{tr}$ and $\mathbf{x}_{te}$.

Remark. The choice of τ_w significantly impacts the precision of contribution tracking. A high τ_w might lead to acknowledgment of fewer clients' contributions. Conversely, a low τ_w might imply equal contribution from all clients. Our empirical findings, outlined in Section VI, show that τ_w should be set to nearly 1.0 for datasets with abundant rules. Yet, for datasets compromised by data poisoning [42], a lower τ_w is advisable to recognize more contributing records in the training set.

Example III.3 (Rule-based Tracing)**.** Recall Figure 2-(b) and examine test instance $(\mathbf{x}_{te}^{(3)}, y_{te}^{(3)} = 0)$ with r_2^- and r_2^- activated, each with weights $w_1^- = 1, w_2^- = 0.5$, respectively. $ct(\mathbf{x}_{te}^{(3)}, y_{te}^{(3)}, \tau_w = 1)$ are training data with the negative label and both r_1^- and r_2^- activated (6 records from participant B). Additionally, the training data with the negative label and only r_1^- activated (2 training records from participant C), are also included by employing a softer threshold of $\tau_w = 0.6$ (*i.e.*, 60% of weighted activated rules), as $\frac{w_1^-}{w_1^- + w_2^-} = \frac{2}{3} \geq 0.6$.

Contribution Allocation. We now present how to compute participants' contributions by allocating credits to the corresponding participants based on the related training data computed using the above tracing strategy. The key issue is to determine the relationship between the amount of related training data and the magnitude of contribution. To this end, we need to identify the differences in effects that various amounts of training data exert on the learned model. According to the classic Federated Averaging (FedAvg) algorithm [43], during training the global model on the data of all participants, the federation computes the aggregated model (weights) update using averaging weighted on the number of training data. Therefore, a larger amount of training data leads to proportionally a larger impact on the learned model. Based on this observation, we propose a **micro contribution allocation scheme** $\phi_v^m(i)$ that distributes the performance gain credits to participants based on the numbers of related training instances:

$$\phi_v^m(i) = \frac{1}{|D_{te}|} \sum_{(\mathbf{x}_{te}, y_{te}) \in D_{te}} \frac{\mathbf{1}[\hat{y} = y_{te}] \cdot |D_i \cap ct(\mathbf{x}_{te}, y_{te}, \tau_w)|}{\sum_{j \in N} |D_j \cap ct(\mathbf{x}_{te}, y_{te}, \tau_w)|} \tag{5}$$

where $\hat{y} = \mathcal{M}_{D_N}(\mathbf{x}_{te})$ is the inference label, and $\mathbf{1}[\hat{y} = y_{te}]$ means we only consider correctly classified test records for contribution allocation, because we use test accuracy as the data metric.

However, there is a potential vulnerability in the micro contribution allocation scheme, as a strategic participant can replicate its data to earn more credits on its matched test instances. This leads to a deficit of other participants that also match these test instances, as the credit allocation is proportional to the amount of related training data. To address this issue, we propose an alternative **macro (or replication-robust) contribution allocation scheme** $\phi_v^M(i)$ that averagely distributes credits to participants with more than a minimally required amount of related training data:

$$\phi_v^M(i) = \frac{1}{|D_{te}|} \sum_{(\mathbf{x}_{te}, y_{te}) \in D_{te}} \frac{\mathbf{1}[\hat{y} = y_{te}] \cdot \mathbf{1}[|D_{related}| \geq \delta]}{\sum_{j \in N} \mathbf{1}[|D_{related}| \geq \delta]} \qquad (6)$$

where $D_{related} = D_i \cap ct(\mathbf{x}_{te}, y_{te}, \tau_w)$, δ is a threshold of minimum related training instances. Intuitively, a higher volume of training data requires a relatively larger δ. Fortunately, we can generate scores for multiple δ values progressively without much extra computation.

Example III.4 (Contribution Allocation)**.** Based on the contribution tracing results with 3 participants and 4 test records shown in Figure 2-(b), we compute micro and macro contribution scores for each participant in Figure 2-(d) and Figure 2-(e), respectively. For instance, participants B and C match the test instance $(\mathbf{x}_{te}^{(3)}, y_{te}^{(3)} = 0)$, with 6 and 2 related training data, respectively. According to the micro scheme, $\phi_v^m(B) = \frac{1}{4} * \frac{6}{6+2} = \frac{3}{16}$ while $\phi_v^m(C) = \frac{1}{4} * \frac{2}{6+2} = \frac{1}{16}$, i.e., proportional to the number of matched training data. According to the macro scheme with $\delta = 2$, $\phi_v^M(B) = \phi_v^M(C) = \frac{1}{4} * \frac{1}{2} = \frac{1}{8}$, i.e., averagely distributed to related participants.

Discussion. We primarily use the micro contribution estimation scheme as the contribution scoring metric, and employ the macro scheme as the auxiliary scheme to identify strategic data replication that results in inflated micro scores (refer to Section IV-A for more details). Notably, computing both of the micro and macro contribution allocation schemes does not add extra computational costs as contribution allocation and rule tracing are independent, and both schemes use the same rule tracing results.

Efficient Computation of CTFL. We have $|D_N|$ rule activation vectors of training data and $|D_{te}|$ rule activation vectors of test data. During the training stage, the overhead is the computation time of a single model learned on D_N. During the model inference phase for contribution allocation, a brute-forth method would involve comparing the rule activation vector of each test instance with the rule activation vectors of all the training data, resulting in $O(|D_{te}| \cdot |D_N|)$ computation overhead. Fortunately, the contribution computation of test instances is not interdependent, and thus the computation process can be fully parallelized with GPUs. However, when the training and test data sets are significantly large, this estimation process is much slower. In such cases, we employ frequent item sets searching algorithms such as Max-Miner [19] to partition the test data into groups, where each group includes test data with the same subset of frequently activated rules. We then compute the related training data for each specific group of test data based on its frequently activated rule subset. Finally, we compute the exact set of related training data for each test instance based on the much smaller-sized related training data belonging to its enclosing group and compute the contribution as usual.

D. *Theoretical Properties of CTFL*

A critical question is that whether CTFL, with the above contribution tracing and allocation techniques, can accurately estimate participant contributions, as the state-of-the-art `ShapleyValue` scheme. Fortunately, CTFL is designed to fulfill the following essential theoretical properties, ensuring fair and rational estimates.

- *Group Rationality.* The sum of all participants' contributions should be equal to the data utility of all participants, *i.e.,* $\sum_{i=1}^{N} \phi_v(i) = v(D_N)$. It can be verified that CTFL satisfies this: $\sum_{i=1}^{N} \phi_v(i) = \frac{1}{|D_{te}|} \sum_{\mathbf{x}_{te}, y_{te}} \mathbf{1}[\hat{y} = y_{te}] = v(D_N)$, where $v(D_N)$ is the test accuracy of the global model. Notably, group rationality can also be applied to other performance metrics such as F-1 score by modifying the allocation formula according to the performance metric.
- *Symmetry.* If two participants bring the same performance gain to a global model, their contributions should be the same. If the performance gain brought by two participants are same, they contribute to the same number of correctly classified test instances, and the contribution allocation of CTFL produces the same contributions.
- *Zero Element.* When the performance gain brought by a participant is zero, its data contribution should also be zero. Specifically, if a participant i does not contribute to classifying any test instance, under the procedure of CTFL, the data contribution attributed to i is zero, *i.e.,* $\phi_v(i) = 0$. This is because participants are unlikely to coincidently hit the rule activation vectors of the test data, since the rules are composed of complicated conjunctive and disjunctive predicates, and there are multiple activated rules to be matched.
- *Additivity*. When utilizing multiple data utility metrics, the contribution are cumulative over these metrics, *i.e.,* $\phi_{u+v}(i) = \phi_u(i) + \phi_v(i)$, where u and v represent two metrics. As CTFL's contribution estimation is based on deterministic allocation formulas, it can compute the contribution including a new metric incrementally, instead of constrained optimizations (*e.g.,* `LeastCore`) where the optimal boundary changes with new constraints.

IV. Robustness and Interpretability

A. *Robustness of CTFL*

In FL, participants are generally assumed to be semi-honest [44]. This assumption implies that while they will cooperate in learning the global model, they may still attempt to inflate their contribution scores to get a larger revenue. For example, participants might attempt to replicate data or include low-quality data, such as poorly labeled (or annotated) records [45]. Regrettably, `ShapleyValue` method [9], [24] cannot address and combat these attempts due to its black-box data utility evaluation methodology. In other words, the data utility of each participant is indirectly evaluated as a whole through the performance metric. Moreover, current black-box approaches are also susceptible to malicious attacks with insignificant contribution fluctuations. Consider the well-known label-flipping attack, in which a malicious participant introduces a small set of label-flipped training data that does not significantly impact the test performance but is detrimental to the deployed FL model [46], [47]. Nevertheless, our proposed method (CTFL) is devised to be robust against these strategic and malicious behaviors by examining the model performance gain and loss caused by each participant.

- *Data Replication.* The macro contribution allocation scheme of CTFL is designed to withstand data replication since clients with related training data of a certain threshold or above receive an identical share of given credits. Specifically, Equation (6) ensures all associated clients to receive the same credit allocation for the test instance, without taking the number of training data beyond a certain threshold into account. This prevents strategic participants from gaining additional benefits. For example, in Figure 2-(d) and (e), the macro scheme allocates participants B and C the same contribution scores, despite that B has 4 more related training instances than C.
- *Low-quality Data.* Low-quality data does not account for contribution credits, because we use a stringent rule-based contribution tracing strategy to identify poorly labeled low-quality data. The utilized logical rules consist of complex predicates featuring various conjunctions and disjunctions. Moreover, to match with a test instance, several interconnected rules must be activated simultaneously. Therefore, it is challenging to manipulate scores by matching multi-dimensional rules coincidentally, especially under a strict strategy with a large τ_w. It is important to note that participants cannot activate all rules to align with the withheld test data, as the potential combinations of rules are exponentially vast, and some rules may be mutually contradictory.
- *Label-flipped Data.* Although the activated rules for label-flipped data may align with corrsponding test records, this alignment fails to account for contribution credits because the associated labels are flipped, *i.e.,* cannot fulfill $\mathbf{1}[\hat{y} = y_{te}]$. Furthermore, by analyzing instances of performance loss (false positives and false negatives), we are able to identify and eliminate such label-flipped data. Specifically, we can trace the loss caused by each participant by modifying the indicator function of Equation (5) from $\mathbf{1}[\hat{y} = y_{te}]$ to $\mathbf{1}[\hat{y} \neq y_{te}]$ (also applies to Equation (6)). This approach can identify potential label-flipping attacks because, in unintentional misclassification test cases, there will be much fewer coincident matching of rule activation vectors with contradictory labels. For example, in Figure 2-(b), normally misclassified test records cannot be aligned with many training data, *e.g.,* false positive $\mathbf{x}_{\mathbf{te}}^{(\mathbf{2})}$, which does not match any training data. However, if a participant such as C has more training instances of label-flipped matches, *e.g.,* $\mathbf{x}_{\mathbf{te}}^{(\mathbf{4})}$, it could be a malicious participant.

Discussion. The ability of adverse clients to manipulate data is limited by their lack of knowledge regarding the test set and the applied rules. Nevertheless, in situations where malicious clients have knowledge of the test set, the applied rules, or the data belonging to other participants, they could potentially create poisoned data. This issue is particularly acute in cases of attack collusion within FL [42], [48]. Tackling these challenges technically exceeds our present capabilities.

B. Interpret Participants' Contributions

We introduce how to interpret participants' beneficial and harmful characteristics by fully exploiting the contribution tracing procedure. Naturally, we can record the rule activation times for each participant during the contribution tracing process, based on which we can identify the high-frequently activated rules to understand each participant's beneficial characteristics. Similarly, we can interpret the harmful characteristics of potential malicious participants, if any, by recording the high-frequently activated rules of each participant that lead to misclassifications. Additionally, we can also determine the useless (or low-quality) data ratio of each participant by recording the matching times of the training data on the test records and reporting the ratio of data that never being matched. Notably, the above rule activation counts are regularized according to the rule weights, *i.e.,* rules with higher weights are more important and should be prioritized.

Example IV.1 (Interpret Participants' Contributions)**.** In Figure 2-(b) and (c), the beneficial characteristics of participant A are summarized by $r_2^+ = \mathit{edu\text{-}years} > 15 \wedge \mathit{work\text{-}class} = \mathit{state\text{-}gov}$, which corresponds to 4 training instances of the positive class. Similarly, the beneficial characteristics of B is summarized by $r_1^- = \mathit{capital\text{-}gain} < 5k$, $r_2^- = \mathit{work\text{-}hours} > 14 \vee \mathit{marital\text{-}status} = \mathit{never}$ which correspond to 6 training instances of the negative class.

Guide Data Collection. There are occasions when the training data fails to effectively cover the test scenarios. Such situations are implied by misclassified test data whose rule activation vectors cannot be matched to a sufficient number of training data. Note this is different from the label-flipped cases with sufficient misleading training data. To improve on these test scenarios, we identify the common patterns of these misclassified cases, so that we can guide participants to collect targeted training samples that cover these scenarios. We achieve this by computing the aggregated rule activation times for misclassified test data and extracting useful patterns from the most frequently activated rules.

V. A Practical Rule-based Model

Equipping a rule-based model as a task model in CTFL raises three significant concerns that must be addressed.

(1) Generalizability. Classic rule-based models, e.g., decision trees, lack sufficient generalizability to handle complicated scenarios. This limitation can result in low performance, leading to inaccurate contribution estimation compared to widely adopted task models in FL, such as deep neural networks.

(2) Precise Tracing Capability. Existing rule-based models normally produce fuzzy rules (*i.e.,* rules that are all partially activated) to improve the model performance. However, these fuzzy rules impede precise tracing of the related training data.

(3) Data Privacy. Rule-based models may need to inspect training data to generate rules (*e.g.,* by computing information gain or Gini impurity [49]), particularly for discretizing

continuous features into intervals. This process violates the data privacy constraints in FL.

To address the above challenges, we propose a practical implementation of the rule-based model, which extends the classical neural network architecture by integrating logical layers. We will first introduce how to encode input features *privately* while capturing their underlying classification patterns. Then, we adopt logical neural networks to build rule-based models with high generalization ability akin to neural networks [25]. Last, to further produce non-fuzzy rules with definite logical predicates for precise contribution tracing and interpretation, we leverage gradient grafting techniques [27] to binarize logical neural weights.

Encode Input Features. Discrete features are encoded into one-hot embeddings, where the value choices of each discrete feature can be fixed by the federation. For example, the number of choices of a discrete feature can be all the unique choices from the federation reserved test dataset plus an unknown flag to denote all other unseen choices that appear in participants' training datasets.

Continuous features need to be discretized to build comprehensible rules. However, we cannot inspect the private training data to figure out reasonable discretization boundaries. On the other hand, simply cutting distribution domain of a continuous feature into uniform bins is not acceptable, because the underlying distribution patterns with respect to the task are overlooked. Thus, the discretization strategy should fully consider the real data distribution of all participants without inspecting the data.

To this end, we build a binarization neutral layer to learn good discretization boundaries in an end-to-end manner. Specifically, for each continuous feature variable $c \in [c_l, c_u]$ to be discretized, we randomly generate lower bounds $l_{1:\tau_d}$ and upper bounds $u_{1:\tau_d}$ that can be combined to generate various discretized intervals, where τ_d is the number of discretization bounds. In this way, we transform c into a binary vector $c_b = [\mathbf{1}(c > l_1), \cdots, \mathbf{1}(c > l_{\tau_d}), \mathbf{1}(u_1 > c), \cdots, \mathbf{1}(u_{\tau_d} > c)]$, where $\mathbf{1}(\cdot)$ is an indicator function. As a consequence, continuous features can be processed like discrete features by applying an extra binarization layer. By learning and discretizing the forward weights on these $2\tau_d$ bounds, we choose appropriate bounds to fit the data distribution of participants for the task without inspecting the data, as there are enough candidate bounds to generate various discretized intervals.

Build Logical Rules. Based on the above encoded features, we are now ready to introduce logical (neural network) layers to derive rules. A logical layer is composed of conjunction nodes and disjunction nodes to produce conjunctive and disjunctive predicates based on logical activation functions:

$$
\begin{aligned}
Conj(\mathbf{x}, \mathbf{w}) &= \prod_i F_c(x_i, w_i), \text{where } F_c(x_i, w_i) = 1 - w_i(1 - x_i) \\
Disj(\mathbf{x}, \mathbf{w}) &= 1 - \prod_i (1 - F_d(x_i, w_i)), \text{where } F_d(x_i, w_i) = x_i \cdot w_i
\end{aligned}
\tag{7}
$$

where $\mathbf{x}$ is the layer input, *i.e.,* an encoded feature vector or the output rule vector from the preceding logical layer, and $\mathbf{w}$ is the logical neural weight vector with $w_i \in [0, 1]$ controlling the logical predicate x_i's involvement degree on the conjunction or disjunction operation. We have $Conj(\mathbf{x}, \mathbf{w}) = \bigwedge_{w_i=1} x_i$ and $Disj(\mathbf{x}, \mathbf{w}) = \bigvee_{w_i=1} x_i$ when $\mathbf{x}$ and $\mathbf{w}$ are both binary vectors. Therefore, we can build compound logical rules via recursive conjunction and disjunction operations with multiple logical layers, where skip connections between logical layers are applied to generate both simple and complicated rules simultaneously. Last but not least, we append a linear layer to accomplish the rule-weighted classification.

Learn Non-fuzzy Rules. So far, we have implemented a rule-based model where the continuous logical weights can be trained by using gradient descent methods. However, the trained weights are fuzzy values in the range of $[0, 1]$, *i.e.,* input logical predicates (or conditions) are all partially activated, rather than desired binary weights which can produce non-fuzzy rules of definite conjunctions and disjunctions. To overcome this issue and produce non-fuzzy rules for precise contribution tracing and interpretation, we now describe how to build binarized logical weights as follows.

Evidently, binarized logical weights are non-differentiable and thus are hard to train directly with gradient decent methods. To solve this issue, we adopt the gradient grafting technique to learn binarized logical weights by leveraging (or grafting) the gradients from the trainable continuous model. Intuitively, in stem grafting, the stems of some plant can be grafted on another plant's root. Similarly, in gradient grafting, we want to graft the training loss of a discrete model $\bar{\mathcal{M}}$ with non-fuzzy rules on the differentiable continuous model $\mathcal{M}$. Specifically, we denote the parameters of $\mathcal{M}$ as θ, and the parameters in training step t as θ^t. We use indicator function $\mathbf{1}(\theta > 0.5)$ to binarize θ to get $\bar{\mathcal{M}}$ with minimum activation threshold 0.5. Let $Y = \mathcal{M}(\theta^t, X)$ and $\bar{Y} = \bar{\mathcal{M}}(\mathbf{1}(\theta^t), X)$ be the continuous and discrete model outputs in step t, respectively. By gradient grafting we update the model parameters as $\theta^{t+1} = \theta^t - \eta \frac{\partial \mathcal{L}(\bar{Y})}{\partial \bar{Y}} \cdot \frac{\partial Y}{\partial \theta^t}$. where η is the learning rate and $\mathcal{L}(\cdot)$ is a loss function to learn $\bar{\mathcal{M}}$, *e.g.,* cross entropy loss. Thus, we learn the discrete model $\bar{\mathcal{M}}$ by combining the gradients of the loss function $\mathcal{L}(\cdot)$ on $\bar{\mathcal{M}}$ and the gradients of the differentiable $\mathcal{M}$. Note that the binarized neural weights of $\bar{\mathcal{M}}$ including the neural weights between an encoding layer and a logical layer, and between logical layers, but not between the last logical layer and the linear layer for weighted classification.

Example V.1 (Rule-based Model)**.** Figure 3 presents a rule-based model using logical neural networks and binarization techniques. Continuous features are firstly discretized via a binarization layer with random lower and upper bounds. Discrete features are encoded into one-hot embeddings and concatenated with the binarization layer output. Two logical layers are followed to learn compound logical rules with conjunctions and disjunctions, *e.g.,* $\wedge_2 : c < u_1 \wedge d = d_1$. Finally, a linear layer outputs the final inference label by aggregating the weighted scores of all activated rules.

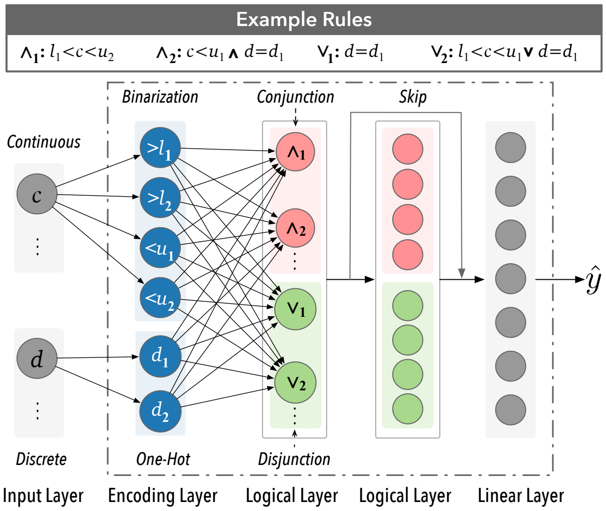


Fig. 3: Rule-based Model using Logical Neural Networks

Data Privacy Analysis. We ensure the fulfillment of the data privacy constraints of FL by maintaining data privacy throughout the entire model training, evaluation, and interpretation process using rule-based models. This implies that we avoid exchanging or inspecting any private data from each participant and do not introduce additional privacy risks. Firstly, during the encoding of input features, the federation itself produces the one-hot encodings of discrete features, while the continuous features are cut into appropriate ranges in an end-to-end model training manner using only value domains and model gradients. Secondly, during the model training phase, the procedure is similar to the original FL global training, except that a rule-based model is used as the task model. Thirdly, during the contribution tracing phase, only the discretized activated rules of the training data are used to align with the test data and compute contribution scores. To ensure privacy, we let participants to generate rule activation vectors of its training data and upload to the federation, which are then used to compute the relevant training data for the test data. This approach is reasonable since the activation vectors encode only task-specific patterns that are useful for model inference, and can be further perturbed to guarantee differential privacy [50]. Lastly, during the interpretation phase, we summarize the aggregated high frequent characteristics based on the recorded rule activation frequency during the preceding contribution tracing phase. Besides, security protection techniques such as secret sharing [51] can also be applied like in regular FL to reduce security risks. Overall, CTFL ensures that data privacy constraints in FL are fulfilled.

VI. Experiments

A. Setup

Datasets. The experiments are conducted on four public classification datasets that meet three crucial requirements: (a) Utilization of common benchmarking datasets from the UCI repository [52] and Kaggle [53]; (b) Inclusion of diverse task performance levels from 50% to 100% test accuracy; (c) Incorporation of multiple social scenarios, such as predicting individual's income level (*adult*), predicting bank term deposits (*bank*), and predicting game winners (*tic-tac-toe* and *dota*2). Table IV shows the primary characteristics of these datasets. To evaluate the performance of CTFL under different FL scenarios, we design the following data distribution cases.

TABLE IV: Datasets

Dataset	#-Instances	#-Features	Feature Type
tic-tac-toe [55]	958	9	discrete
adult [56]	32, 561	14	mixed
bank [57]	45, 211	16	mixed
*dota*2 [58]	102, 944	116	discrete

- *Skew sample.* Participants have varying amounts of data from the same distribution. The training data is randomly partitioned and distributed to all participants. To control the skewness of data ratios, we randomly sample participants' data ratios from the Dirichlet distribution, which generates ratios that sum up to 1.0. The degree of the skewness of data ratios is controlled by a hyper-parameter α.
- *Skew label.* Participants have varying amounts of data with different label distributions. The data in each class label is randomly partitioned and distributed to participants, where the data ratios of each label are separately sampled from the Dirichlet distribution.
- *Data replication* [54]. We randomly select some participants from the above skew-label data, and replicate their data by a random data ratio.
- *Low-quality data.* We randomly select some participants from the above skew-label data, sample their training data by a random data ratio, and modify the class labels randomly according to the participant's label distribution.
- *Label-flipped data.* We randomly select some participants from the above skew-label data, sample their training data by a random data ratio, and flip the class labels of these data.

Baselines. We compare CTFL with the following four kinds of contribution evaluation schemes.

- `Individual`: $\phi_v(i) = v(D_i), i \in N$. We consider a typical individual scheme that regards a participant's learned model inference accuracy as its contribution [21].
- `LeaveOneOut`: $\phi_v(i) = v(D_N) - v(D_{N\setminus\{i\}}), \forall i \in N$. `LeaveOneOut` regard the performance loss of removing a participant as its contribution [22].
- `ShapleyValue`: $\phi_v(i) = \mathbb{E}_{S\subseteq N\setminus i}[v(D_{S\cup\{i\}}) - v(D_S)]$ $, \forall i \in N$. `ShapleyValue` regards the expected marginal performance gain brought by a participant as the contribution [9]. Notably, $\Theta(n^2 \log n)$ participant combinations are sampled to speed up the computation and guided sampling and early stop during training are also applied, according to [9].
- `LeastCore`: $\phi_v(i)$, that fulfills Equation (2). Least core aims to minimize the score deficit of each potential coalition [23]. We sample and compute $\Theta(n^2 \log n)$ linear constraints to speed up the constraints and linear programming computation, according to [23].

Remark. Methods aimed at offering model interpretation, such as perturbation-based techniques [59], simulate perturbed samples in the local neighbourhood of a data point (sample) to interpret how a black-box model is affected by this data point rather than to predict the contribution score of data records held by a client. Therefore, these approaches are not directly applicable to contribution estimation and individual contribution tracing in the context of federated learning.

Metrics. We evaluate the accuracy, efficiency, robustness and interpretability of contribution estimation.

- *Remove High-contribution Participants.* We eliminate participants with the top five contribution scores one by one in descending order (without replacement), train the model on the remaining data after each elimination, and compute the model test performance. When a method's contribution estimation is more accurate, removing participants with top contributions affects the learned model performance more significantly, leading to a faster drop in accuracy, *i.e.,the smaller the area under the model accuracy curve, the better*.
- *Execution Time.* We evaluate the execution time of each method for producing the final contribution scores.
- *Robustness.* We evaluate the relative contribution score changes of the corresponding participants after data replication, injecting low-quality samples and label-flipped samples, *i.e.,* $\frac{\phi_v(i')-\phi_v(i)}{\phi_v(i)}$, where i' represents i after modifying the data. We expect a robust method to report *near-to-zero* contribution fluctuation after data replication, and report contribution loss that is proportional to the amount of injected low-quality or label-flipped data.
- *Interpretability.* We conduct a case study on datasets *tic-tac-toe* and *adult*, and analyze whether the generated participants' characteristics are comprehensible and insightful.

Default Parameters. The number of FL participants is set to 8, as it's impossible to execute ShapleyValue and LeastCore in a reasonable time (*i.e.,* 24 hours) with more participants (*e.g.,* already more than 10 hours to run accelerated ShapleyValue in current settings). Furthermore, the number of updated participants in robustness scenarios is set to 2. The activation threshold for rule tracing is in the range of $[0.8, 1]$. The dimension of binarization layer is set to 10. The number of logic layers is set to 1 with the number of dimensions in the range of $[64, 512]$. α for Dirichlet distribution is set in $[0.6, 1]$. The sampled data ratios for robust data scenarios vary uniformly in $[0.1, 0.5]$. All experiments are repeated 10 times to compute the average results.

Environment. All methods are implemented in Python 3.10 and evaluated on a Linux server with Intel 3.10GHz CPU and 256GB memory, and RTX 3090 GPU with 24GB memory and CUDA 11.4. Please also refer to our code at https://github.com/knifelee/ctfl.

B. Quantitative Evaluation

We evaluate the performance of CTFL compared to baselines and specifically answer the following research questions.

RQ1. Is CTFL capable of estimating the contribution of participants accurately? The most important goal in contribution estimation is contribution ranking accuracy. Thus, we first evaluate whether CTFL can perform accurately as the theoretically optimal ShapleyValue scheme and outperform other methods. Figure 4 shows the results. ShapleyValue and LeastCore do not appear in the largest dataset *dota2*, because they cannot finish in a reasonable running time. We make the following observations.

1) CTFL achieves state-of-the-art contribution estimation accuracy, and even performs better than the approximated ShapleyValue. CTFL occupies the bottom of all subfigures, outperforms all existing methods in *tic-tac-toe* and *adult* datasets with skew-label data distributions, and performs similarly to the best baseline in other cases. This is because the rule-based contribution tracing of CTFL is effective on estimating participants' contributions.

2) Skew-label cases, where participants have different data label distributions, are harder to estimate than skew-sample cases. Our CTFL performs stably on both types of cases. In skew-label cases, the contributions are more skewed among participants compared to skew-sample cases. The contribution allocation of CTFL is precisely associated with individual data records and is not influenced.

3) CTFL$_{\text{micro}}$ and CTFL$_{\text{macro}}$ performs similar in datasets with high task performance, and CTFL$_{\text{micro}}$ performs better otherwise. They performs similar on *tic-tac-toe*, *adult* and *bank* with relatively higher performance, while CTFL$_{\text{micro}}$ performs much better than CTFL$_{\text{macro}}$ on *dota2* with lower performance. This is because the high-quality training data in low-performance *dota2* is still insufficient, where the contribution is still proportional to the number of scarce high-quality training samples (which can be confirmed by Individual's good performance on *dota2*).

4) Baselines except ShapleyValue perform badly on estimating participants' contributions. This is because although LeastCore achieves equilibrium from a coalition perspective, it weakens individual's benefits and thus performs worse. And Individual neglects the value of cooperation, while LeaveOneOut neglects the contribution of participants with homogenous data, and can only accurately estimate the one whose leaving causes the maximum deficit.

RQ2. Is CTFL more efficient than existing approaches? We evaluate whether CTFL with only a single pass of model training and contribution tracing is more efficient than baselines with multiple times of model training and inference. Based on the results in Figure 5, we make the following observations.

1) CTFL is a very efficient method compared to existing approaches. In particular, CTFL is 2-3 orders of magnitude faster than widely adopted methods ShapleyValue and LeastCore. This is because CTFL only train and evaluate a single task model, while other approaches need to train and evaluate multiple models to compute participant contributions. Besides, we speed up computation by finding frequent rule activation sets to reduce computation.

2) The execution time of CTFL is similar to Individual. This is because both methods compute backward gradients for

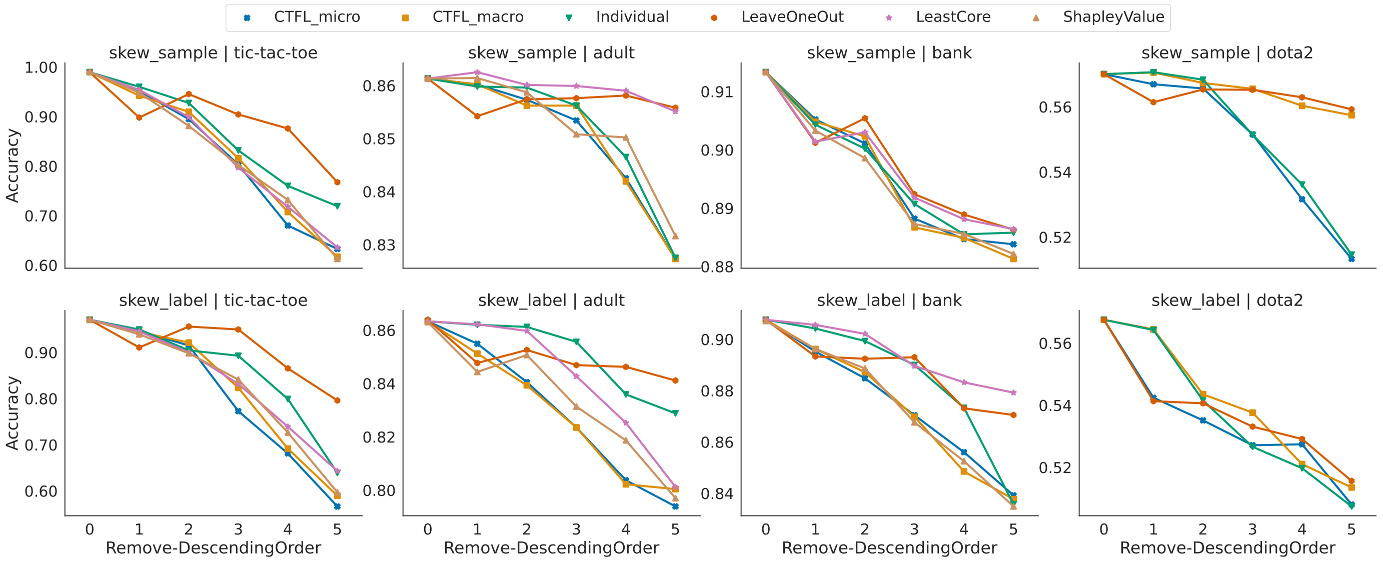


Fig. 4: Accuracy by Removing Participants in Contribution Descending Order (the smaller the area under curve, the better)

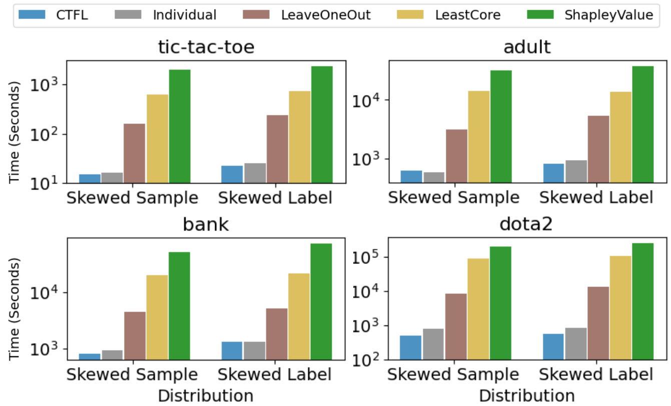


Fig. 5: Execution Time.

the same number of data records, and Individual takes extra time to update model parameters for n models, while CTFL traces the contribution credits for all participants.

3) ShapleyValue and LeastCore are too slow to be applied in real-world scenarios, and take almost one day to estimate participants' contributions on datasets with tens of thousands of records, which is dozens even hundreds times of the original FL model training time. Thus, it is hard to apply them due to high time and computation resource costs. The implementations of these two methods in this paper are based on accelerated techniques such as sampling and early stop.

4) Upon comparing outcomes across various datasets, it was noted that there is a direct relationship between execution time and the feature dimensionality. This correlation arises from the fact that the computation of gradients during global model training is dependent on the quantity of features.

RQ3. Is CTFL more robust to common adverse behaviors? We evaluate whether the contribution scores of CTFL can react appropriately to the adverse behaviors including data replication, low-quality records and label-flipped data, and Figure 6 shows the results respectively in three rows. The relative change values are clipped to be in $[-1, 1]$. We make the following observations.

1) $\text{CTFL}_{\text{macro}}$ and Individual are robust to data replication. The contribution scores of $\text{CTFL}_{\text{macro}}$ and Individual do not increase after participants' replicating their data, *i.e.,* the score changes are near to zero. For $\text{CTFL}_{\text{macro}}$, this is because the contribution volume does not depend on the number of matched training data after fulfilling a threshold. For Individual, this is because participants' contributions are evaluated separately.

2) $\text{CTFL}_{\text{micro}}$ and Individual are robust to low-quality and label-flipped data. The contribution scores of $\text{CTFL}_{\text{micro}}$ and Individual show proportional and stable reduction after participants injecting low-quality or label-flipped data. This is because the contribution computation of $\text{CTFL}_{\text{micro}}$ is precisely according to the number of beneficial training data provided by each participant. And for Individual, the adverse behaviors severely affects model performance as each participant's training data are evaluated separately.

3) LeaveOneOut, LeastCore and ShapleyValue do not react appropriately or stably to all three types of adverse behaviors due to duplicated model training on different coalitions of clients.

C. Case Study on Contribution Interpretation

We conduct a case study on datasets *tic-tac-toe* and *adult* with the skew-label scenario including three participants to show that CTFL can interpret participants' characteristics.

Interpret for *tic-tac-toe*. The *tic-tac-toe* dataset is about a two-player game whose play rules is demonstrated in Figure 7 (a), where we denote the winning of player x by $+$ and o win by $-$. We collect the frequently activated rules of three participants that are beneficial to classifying test data (shown in Figure 7-(b)), where superscript $+$ of rules means supporting x to win and $-$ for supporting o. We extract the following characteristics from these rules.

TABLE V: Selected Frequently Activated Rules of Three Participants (A/B/C) on *Adult*, +/- Refers to Positive/Negative Class

#	A	B	C
1	capital-gain $<$ 5k$^-$	capital-gain $<$ 5262$^-$	marital-status = never $\wedge$ hours-per-week $>$ 14$^-$
2	capital-gain $<$ 5k $\wedge$ capital-loss $<$ 1k$^-$	capital-gain $<$ 5k $\wedge$ capital-loss $<$ 1k $\wedge$ age $<$ 33$^-$	work-class = private or state-gov $\vee$ age $>$ 55$^+$
3	fnlwgt>43800 $\wedge$ capital-gain<5k$^-$	fnlwgt $>$ 44k $\wedge$ capital-loss<1k$^+$	capital-gain $>$ 21k $\vee$ education-num $>$ 15$^+$

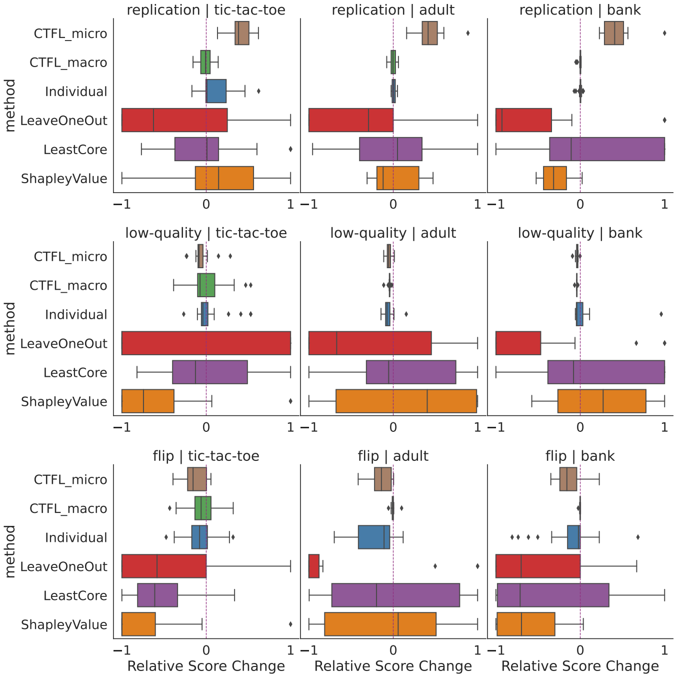
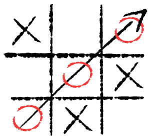


Fig. 6: Contribution Estimation Robustness Evaluation



#	A	B	C
1	$1_x \wedge 2_x \wedge 3_x^+$	$4_x \wedge 5_x \wedge 6_x^+$	$7_o \wedge 8_o \wedge 9_o^-$
2	$3_x \wedge 6_x \wedge 9_x^+$	$2_x \wedge 5_x \wedge 8_x^+$	$1_o \wedge 6_o^-$
3	$7_x \wedge 8_x \wedge 9_x^+$	$1_x \wedge 2_x \wedge 3_x^+$	$4_x \wedge 5_x \wedge 6_x^+$

Fig. 7: *Tic-tac-toe*: (a) The player o wins with $3_o \wedge 5_0 \wedge 7_o$; (b) Selected frequently activated rules on three participants.

- Participant A and B mainly hold data records on x's winning cases, and they share one typical case $1_x \wedge 2_x \wedge 3_x$.
- Participant C mainly holds data records on o's winning cases, and also holds some winning records on x, *i.e.,* $4_x \wedge 5_x \wedge 6_x$.
- A short rule can also be useful, *e.g.,* o is more likely to win when we have $1_o \wedge 6_o$.

Interpret for ***Adult*****.** The *adult* dataset predicts adult's income levels as low (income$\leq 50k\$$, denoted by $-$) or high (income $> 50k\$$, denoted by $+$). The selected frequently activated rules of three participants are shown in Figure V, based on which we make the following observations.

- Low-income data are dominant in all three players. The frequent rules of all players mainly support the negative class, *i.e.,* low income, as low income is the dominant label in the *adult* dataset.
- The data of participants A and B are homogeneous. A and B share similar negative activated rules with the same predicates, *e.g., capital-gain* $< 5k$ and *capital-loss* $< 1k$.
- Participant C holds high-income data records. C has more frequently activated rules associated with the high-income label, *e.g., age* > 55 and *education-num* > 15.

Summary. We summarize the following important findings.

1) CTFL estimates participants' contributions accurately as the theoretically optimal `ShapleyValue`, while the computation time of CTFL is 2-3 orders of magnitude less than `ShapleyValue`.
2) CTFL is robust to common adverse behaviors including data replication, low-quality data and label-flipping attacks. Contribution credits by CTFL remain consistent and stable in response to these adverse behaviors. In contrast, `LeaveOneOut`, `LeastCore` and `ShapleyValue` are not robust to these adverse behaviors and their contribution credits fluctuate sharply. Besides, although `Individual` is robust to these adverse behaviors, it cannot effectively estimate each participant's cooperative contribution.
3) CTFL is able to interpret each participant's contribution credits and provide contribution allocation evidence and insightful characteristics into each participant, which help the federation and participants to understand their roles.

VII. Conclusion

In this paper, we introduce CTFL, an innovative framework for estimating participants' contributions in FL. Our approach efficiently and accurately assesses participants' contributions by tracking the test performance gain through rule-based models. Furthermore, we establish that CTFL ensures rational estimations by satisfying essential theoretical properties. The incorporation of carefully designed contribution allocation schemes enhances the robustness of CTFL. We also provide interpretability to participants' contribution scores, offering insightful properties for each participant. Besides, we present a practical rule-based model utilizing logical neural networks and binarization techniques. Experimental results demonstrate that CTFL outperforms baselines significantly across accuracy, efficiency, robustness, and interpretability. The promising performance exhibited by CTFL underscores its potential as a versatile framework for contribution estimation in FL settings. Future research directions involve extending CTFL to support vertical federated learning and devising a systematic incentive mechanism leveraging the capabilities of CTFL. Furthermore, extending CTFL to facilitate interpretable contribution estimation for federated learning in image and text classification is a promising future direction, while extra efforts from domain experts in rule extraction or pattern design should be taken into consideration in such contexts.

Acknowledgement. This paper was supported by National Key R&D Program of China under Grant Number 2023YFB4503600, NSF of China (61925205, 62232009, 62102215), and Zhongguancun Lab.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[2] F. Fu, H. Xue, Y. Cheng, Y. Tao, and B. Cui, "Blindfl: Vertical federated machine learning without peeking into your data," in *Proceedings of the 2022 International Conference on Management of Data*, 2022, pp. 1316–1330.

[3] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[4] R. C. Fernandez, "Data-sharing markets: Model, protocol, and algorithms to incentivize the formation of data-sharing consortia," in *Proceedings of the 2023 ACM SIGMOD international conference on Management of data*, 2023.

[5] R. Fu, Y. Wu, Q. Xu, and M. Zhang, "Feast: A communication-efficient federated feature selection framework for relational data," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–28, 2023.

[6] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn *et al.*, "Ibm federated learning: an enterprise framework white paper v0. 1," *arXiv preprint arXiv:2007.10987*, 2020.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[8] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.

[9] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, "Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–21, 2022.

[10] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and S. Avestimehr, "Federated learning for internet of things: Applications, challenges, and opportunities," *arXiv preprint arXiv:2111.07494*, 2021.

[11] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gürel, B. Li, C. Zhang, C. J. Spanos, and D. Song, "Efficient task-specific data valuation for nearest neighbor algorithms," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1610–1623, 2019.

[12] L. S. Shapley, "A value for n-person games," *Annals of Mathematical Studies*, vol. 28, pp. 307–317, 1953.

[13] A. Ghorbani, M. Kim, and J. Zou, "A distributional framework for data valuation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3535–3544.

[14] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos, "Towards efficient data valuation based on the shapley value," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1167–1176.

[15] R. Mitchell, J. Cooper, E. Frank, and G. Holmes, "Sampling permutations for shapley value estimation," 2022.

[16] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and S. Y. Philip, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE transactions on neural networks and learning systems*, 2022.

[17] X. Tu, K. Zhu, N. C. Luong, D. Niyato, Y. Zhang, and J. Li, "Incentive mechanisms for federated learning: From economic and game theoretic perspective," *IEEE Transactions on Cognitive Communications and Networking*, 2022.

[18] M. Kuhn, K. Johnson, M. Kuhn, and K. Johnson, "Classification trees and rule-based models," *Applied predictive modeling*, pp. 369–413, 2013.

[19] R. J. Bayardo Jr, "Efficiently mining long patterns from databases," in *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, 1998, pp. 85–93.

[20] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y.-C. Liang, "Joint service pricing and cooperative relay communication for federated learning," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2019, pp. 815–820.

[21] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[22] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2597–2604.

[23] T. Yan and A. D. Procaccia, "If you like shapley then you'll love the core," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 6, 2021, pp. 5751–5759.

[24] T. van Campen, H. Hamers, B. Husslage, and R. Lindelauf, "A new approximation method for the shapley value applied to the wtc 9/11 terrorist attack," *Social Network Analysis and Mining*, vol. 8, no. 1, pp. 1–12, 2018.

[25] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma *et al.*, "Logical neural networks," *arXiv preprint arXiv:2006.13155*, 2020.

[26] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[27] Z. Wang, W. Zhang, N. Liu, and J. Wang, "Scalable rule-based representation learning for interpretable classification," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 479–30 491, 2021.

[28] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, and Z. Shen, "Dealing with label quality disparity in federated learning," in *Federated Learning*. Springer, 2020, pp. 108–121.

[29] J. Lin, M. Du, and J. Liu, "Free-riders in federated learning: Attacks and defenses," *arXiv preprint arXiv:1911.12560*, 2019.

[30] H. Lv, Z. Zheng, T. Luo, F. Wu, S. Tang, L. Hua, R. Jia, and C. Lv, "Data-free evaluation of user contributions in federated learning," in *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. IEEE, 2021, pp. 1–8.

[31] S. K. Shyn, D. Kim, and K. Kim, "Fedccea: A practical approach of client contribution evaluation for federated learning," *arXiv preprint arXiv:2106.02310*, 2021.

[32] J. Zhao, X. Zhu, J. Wang, and J. Xiao, "Efficient client contribution evaluation for horizontal federated learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3060–3064.

[33] J. Yoon, S. Arik, and T. Pfister, "Data valuation using reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 842–10 851.

[34] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*. PMLR, 2017, pp. 1885–1894.

[35] A. Richardson, A. Filos-Ratsikas, and B. Faltings, "Rewarding high-quality data via influence functions," *arXiv preprint arXiv:1908.11598*, 2019.

[36] M. Kearns and D. Ron, "Algorithmic stability and sanity-check bounds for leave-one-out cross-validation," *Neural computation*, vol. 11, no. 6, pp. 1427–1453, 1999.

[37] P. Dubey, "On the uniqueness of the shapley value," *International Journal of Game Theory*, vol. 4, no. 3, pp. 131–139, 1975.

[38] B. Peleg and P. Sudhölter, *Introduction to the theory of cooperative games*. Springer Science & Business Media, 2007, vol. 34.

[39] L. G. Telser, "The usefulness of core theory in economics," *Journal of Economic Perspectives*, vol. 8, no. 2, pp. 151–164, 1994.

[40] T. Hailesilassie, "Rule extraction algorithm for deep neural networks: A review," *arXiv preprint arXiv:1610.05267*, 2016.

[41] A. Jurek, Y. Bi, S. Wu, and C. Nugent, "A survey of commonly used ensemble-based classification techniques," *The Knowledge Engineering Review*, vol. 29, no. 5, pp. 551–581, 2014.

[42] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

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

[44] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[45] L. K. Wang Yong, Li Guoliang, "Contribution evaluation for federated learning: A survey," *Journal of Software(in Chinese)*, vol. 34, no. 3, 2023.

[46] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[47] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[48] S. Tan, F. Hao, T. Gu, L. Li, and M. Liu, "Collusive model poisoning attack in decentralized federated learning," *IEEE Transactions on Industrial Informatics*, 2023.

[49] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, pp. 77–93, 2004.

[50] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[51] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[52] "Uci machine learning repository." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/

[53] "Kaggle." [Online]. Available: https://www.kaggle.com/

[54] X. Xu, Z. Wu, C. S. Foo, and B. K. H. Low, "Validation free and replication robust volume-based data valuation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[55] "Tic-tac-toe endgame data set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame

[56] "Adult income dataset." [Online]. Available: https://www.kaggle.com/datasets/wenruliu/adult-income-dataset

[57] "Bank marketing uci." [Online]. Available: https://www.kaggle.com/c/bank-marketing-uci

[58] "Dota2 games results." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results

[59] S. Agarwal, S. Jabbari, C. Agarwal, S. Upadhyay, S. Wu, and H. Lakkaraju, "Towards the unification and robustness of perturbation and gradient based explanations," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 110–119. [Online]. Available: http://proceedings.mlr.press/v139/agarwal21c.html