LR-XFL: Logical Reasoning-Based Explainable Federated Learning

Yanci Zhang, Han Yu

School of Computer Science and Engineering, Nanyang Technological University, Singapore
Joint NTU-WeBank Research Centre on Fintech, Nanyang Technological University, Singapore
{yanci001, han.yu}@ntu.edu.sg

Abstract

Federated learning (FL) is an emerging approach for training machine learning models collaboratively while preserving data privacy. The need for privacy protection makes it difficult for FL models to achieve global transparency and explainability. To address this limitation, we incorporate logic-based explanations into FL by proposing the Logical Reasoning-based eXplainable Federated Learning (LR-XFL) approach. Under LR-XFL, FL clients create local logic rules based on their local data and send them, along with model updates, to the FL server. The FL server connects the local logic rules through a proper logical connector that is derived based on properties of client data, without requiring access to the raw data. In addition, the server also aggregates the local model updates with weight values determined by the quality of the clients' local data as reflected by their uploaded logic rules. The results show that LR-XFL outperforms the most relevant baseline by 1.19%, 5.81% and 5.41% in terms of classification accuracy, rule accuracy and rule fidelity, respectively. The explicit rule evaluation and expression under LR-XFL enable human experts to validate and correct the rules on the server side, hence improving the global FL model's robustness to errors. It has the potential to enhance the transparency of FL models for areas like healthcare and finance where both data privacy and explainability are important.

Introduction

Federated learning (FL) (Kairouz, McMahan et al. 2021) is a collaborative training paradigm that jointly trains artificial intelligence (AI) models from a set of FL clients without exposing their local data. Under the guidance of a central server, clients improve their local models from the information gained from other clients. The general idea is that clients upload their model updates rather than data to the FL server. The server aggregates the received models and returns the updated global model to clients so that they can further train it. This privacy-protection joint training schema, however, introduces complexities, particularly in explaining the decision-making process, given the server's lack of direct access to client datasets.

Explainable AI (XAI) (Gunning et al. 2019; Xu et al. 2019; Yu et al. 2014) is gaining attention in recent years. The aim of XAI is to make AI model behaviours interpretable to humans by providing explanations. A wide range of studies have successfully provided explanations for black-box AI models. For instance, approaches like Grad-CAM (Selvaraju et al. 2017) analyse the output of the final convolutional layer of a given neural network and calculate how changes to each region of an input image affect the output to identify the most important image regions for decision-making. Model-agnostic algorithms like SHAP (Lundberg and Lee 2017) and LIME (Ribeiro, Singh, and Guestrin 2016) assess the importance of features at the instance level. However, these approaches cannot clearly illustrate the model decision-making processes, but rather provide a post-hoc analysis of the models.

Concept-based models, as a bridge between symbolic AI (based on rules and logic) and statistical AI (e.g., neural networks), extract concepts from the model by mapping the hidden information of the last layer of neurons in a neural network to human-understandable concepts (Kim et al. 2018) or constraining concepts as a part of a neural network (Koh et al. 2020). However, the concepts are still isolated and at the instance level. Logic rules, however, can connect the activated concepts during the prediction to illustrate the reasoning process (Lee et al. 2022). Instance-level logic rules can be further connected to form a global class-level rule (Barbiero et al. 2022). Such logic-based explanations are beneficial as they illustrate the decision-making process in a manner that is readily understood by humans.

Traditional logic-based concept models are designed for centralised AI frameworks and cannot be directly applied in FL settings. Integrating logic rules into FL faces three important challenges:

1. **Local Accuracy vs. Global Representativeness**: Clients, with access to only partial or potentially skewed data, may derive rules that seem accurate within their local context but could introduce biases or misrepresent the broader global perspective. It is imperative for the server to generate comprehensive and globally accurate rules.
2. **Conflict Resolution and Rule Combination**: The merging of logic rules at a central server can lead to conflicts. A simplistic connection of local rules using the logical 'AND' operator might introduce conflicts, and combining rules with the 'OR' operator could dilute the overall rule accuracy. Moreover, determining the optimal com-

bination of rules remains a formidable challenge. Global rules should selectively incorporate accurate rules, using appropriate logical connectors.

3. **FL Client Weight Assignment**: Assigning suitable weights to client model updates during aggregation based on their logic rules presents another challenge.

To tackle these challenges, we propose the Logical Reasoning-based eXplainable Federated Learning (LR-XFL) approach. Under LR-XFL, FL clients create local logic rules based on their local data and send them, along with model updates, to the FL server. The FL server connects the local logic rules through a logical connector (AND or OR) that is adaptively determined by LR-XFL based on properties of client data, without exposing raw data. In addition, the server also aggregates the local model updates with weight values determined by the quality of the clients' local data as reflected by their uploaded logic rules. To the best of our knowledge, LR-XFL is the first FL reasoning approach capable of adaptively aggregating local rules from clients.

To evaluate the performance of LR-XFL, we conduct extensive experiments on four benchmark datasets under FL settings[1]. Compared to three prevailing related alternative approaches, LR-XFL has demonstrated significant advantages. It outperforms the most relevant baseline by 1.19%, 5.81% and 5.41% in terms of classification accuracy, rule accuracy and rule fidelity, respectively. The explicit rule evaluation and expression under LR-XFL enables human experts to validate and correct the rules on the server side, hence improving the global FL model's robustness to errors. It has the potential to enhance the transparency of FL models for areas like healthcare and finance where both data privacy and explainability are important.

Related Work

Concept-based XAI

Concept-based XAI focuses on identifying high-level abstractions or "concepts" in data. In traditional deep learning models, the lower layers often detect edges or textures, while the upper layers detect more abstract features like shapes or objects. Concept-based learning regards the abstract features as "concepts" and aims to make them more explicit. The concepts are often obtained by linking the hidden information of the last layer of a neural network to human-understandable concepts (Kim et al. 2018; Kazhdan et al. 2020), or constraining the structure of the neural network to learn the concepts (Chen, Bei, and Rudin 2020; Ciravegna et al. 2020; Koh et al. 2020; Stammer, Schramowski, and Kersting 2021; Barbiero et al. 2022). The design of a concept-based model makes it suitable to provide explanations for the decision-making process as the concepts are intuitive and understandable for humans, thereby making the decision process transparent. For instance, if a model trained on animal images learns the concept of "wings" and "beaks", it can explain a classification decision by reasoning that the presence of wings in an image is a strong indicator for the "bird" category.

Logic rules serve as a means to link extracted concepts, enabling the decisions made by the model to be explained based on the concepts it incorporates. Logic rules are inherently straightforward and deterministic. Traditional rule-based systems, such as Decision Trees (DTs) (Quinlan 1986), offer intuitive explanations through logic rules. In DTs, each decision path can be interpreted as a rule. Nonetheless, including all features in a decision path can render a rule excessively lengthy with unnecessary features, and consequently diminish its comprehensibility. In the domain of natural language processing, logical reasoning has been adopted for tasks like sentiment analysis (Lee et al. 2022) and text prediction (Jain et al. 2022). For image and tabular data, a novel entropy-based linear layer has been introduced to produce rule-based explanations (Barbiero et al. 2022). However, these approaches require direct access to the training data, making them unsuitable for operations under FL settings.

Federated Learning

FL offers a decentralised and privacy-preserving approach to training AI models on distributedly owned data (Kairouz, McMahan et al. 2021). Instead of moving potentially sensitive raw data, only model updates are being sent back and forth between the FL server and the FL clients. However, such a training approach makes it challenging for explaining the decision-making rationale behind the resulting FL model. The FL field has recognised the potential for logic rules to address this challenge. There have been several notable attempts to integrate logic rules into FL.

In (An and Ma 2023), signal temporal logic is employed to discern the properties of client devices, which are subsequently leveraged to cluster them during model aggregation to produce personalised FL models. Similarly in (Cha et al. 2022), fuzzy logic is used to enhance client selection in vehicular networks. Besides, in (Zhu et al. 2021), the Takagi–Sugeno fuzzy rule is integrated into FL for federated fuzzy clustering. Nevertheless, these approaches are predominantly focused on FL client selection and clustering. They are not designed to address the challenges of building explainable FL with logical reasoning. The proposed LR-XFL approach bridges this important gap.

Preliminaries

The base model under the proposed LR-XFL approach is the entropy-based logic explanations of neural networks (Barbiero et al. 2022). This model adeptly extracts logic-based explanations from neural networks, representing them in logic rules. Designed to handle both images and tabular data, the model provides classification together with rule-based explanations. For image data, it first employs the ResNet10 (He et al. 2016) image processing network to map a given image from the pixel space to the concept space. For tabular data, the mapping can be achieved by linear models. Subsequently, the concept space embedding is mapped to the target class using the entropy-based linear layer. The

[1]The code is available at https://github.com/Yanci87/LR-XFL.

activated concepts are then derived from the parameters of the entropy-based layer to form logic rules.

The process of obtaining logic rules from the entropy-based linear layer relies on a truth table, denoted as T_c, corresponding to class c. This truth table T_c captures the behaviour of the neural network by leveraging Boolean-like representations of the input concepts. Specifically, each row of T_c encompasses activated concepts for a sample predicted to be under class c. The activation status of these concepts for a single data point is determined by processing its concept vector via a binary mask derived from the entropy-based layer parameters. Concepts that are activated through the binary mask for the prediction of class c are included, while others are excluded. Given an activated concept f_i, its representation will either be f_i or $\neg f_i$, depending on its value within the input data point. For every row in the truth table, a sample-level rule-based explanation is formulated by connecting all activated concepts using the AND operator. The class-level explanation for class c is obtained by connecting these sample-level rules, pertaining to data classified under class c, with the OR operator.

However, always using the OR operator to connect sample-level rules can inadvertently make the rule less precise. Consider two sample-level rules: 1) $hasBeak \leftrightarrow Bird$, and 2) $hasWing \leftrightarrow Bird$. Combining these rules with the OR operator yields: $hasBeak \vee hasWing \leftrightarrow Bird$. However, this aggregated rule incorrectly classifies samples only with beaks and no wings, or only with wings and no beaks as birds. In this case, the AND operator is more suitable since the two sample-level rules contain complementary information from different perspectives. The proposed LR−XFL approach is designed to deal with such challenging situations.

The Proposed LR−XFL Approach

In this section, we describe the detailed design of LR−XFL (Figure 1), a first-of-its-kind logic-based explainable framework designed for FL settings built upon the entropy-based network (Barbiero et al. 2022). It consists of a novel method for automatically determining the appropriate logical connector (AND or OR). In addition, it addresses the challenge of rule aggregation when selecting and merging client-generated rules. During FL model aggregation, LR−XFL computes client weights based on their respective rules to support weighted averaging.

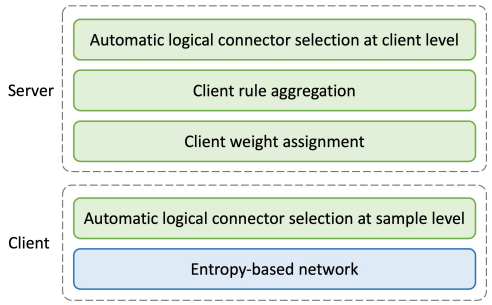


Figure 1: Overall design of LR−XFL.

Algorithm 1: LR−XFL

Input: K clients, each holding a set of local data; a server, holding a set of data for logic validation and testing
Output: Global logic rules for the server; local models and logic rules for clients

while Global model has not achieved the target performance on the validation set **and** max training rounds have not been reached **do**
  For each FL client $k, k \in \{1, \ldots, K\}$:
  Trains the local model;
  Selects the appropriate logical connector for the local rules;
  Generates logic rules r_k^c for $c \in \{1, \ldots, C\}$ classes;
  Uploads the local model and logic rules to the FL server;
  FL Server:
  Selects the appropriate global logical connector based on clients' uploaded logic rules;
  Selects and aggregates clients' local rules;
  Calculates and assigns weights $\{w_1, \ldots, w_K\}$ for the clients based on the performance of their logic rules;
  Aggregates the local models into the global model based on the assigned weights;
  Sends the global model to the clients;
  For each FL client k: Receives the global model and continues training for the next round;
end while

Overview of LR−XFL

Figure 2 depicts the architecture and workflow of LR−XFL. On the client side, entropy-based networks in (Barbiero et al. 2022) are adopted as the base model. Each FL client possesses a set of rules derived from this entropy-based network. The FL server derives a set of global rules from local rules, and is in charge of sending the global model back to clients. Algorithm 1 provides an overview of LR−XFL. Under LR−XFL, FL clients formulate local logic rules grounded in their individual datasets. These rules, along with model updates, are then transmitted to the FL server. Notably, the FL server aggregates the local logic rules through an appropriate connector ($\wedge$ or $\vee$), which is determined based on the characteristics inherent in the client's data without accessing the raw data. Furthermore, the server aggregates local model updates, assigning weight values based on the quality of the clients' local data as gauged from their local logic rules. This iterative training process is carried out until either a predefined maximum number of iterations T is reached, or the global model attains its target performance on the server-side validation dataset.

Determining OR or AND Logical Connector

Ideally, the global rules shall only include the right rules with the correct logical connector. When integrating local logic rules, the first step is to determine whether to use the AND or the OR logical connector. Connecting local rules with AND might lead to conflicts, while introducing partial or wrong rules with OR might undermine accuracy.

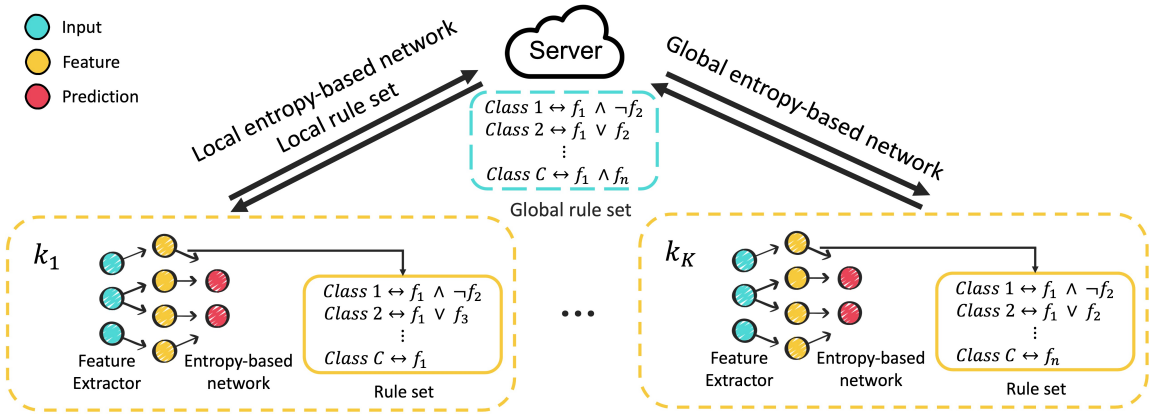


Figure 2: The system architecture and workflow of LR−XFL.

We posit that the underlying rationale for choosing between AND and OR hinges on the potential conflict among the clients' local features. If all features are mutually exclusive (i.e., they cannot coexist in a single data point), the $\vee$ connector shall be used. An example of this scenario is the classification sequence $1 \vee 3 \vee 5 \vee 7 \vee 9$, which denotes the 'odd number' category. On the other hand, in instances where features do not inherently conflict and can appear together in a single data point, the $\wedge$ connector is preferable. For instance, in an animal classification task, features "wings" and "beaks" can co-exist in the "bird" category. Therefore, a rule such as $Wings \wedge Beaks \leftrightarrow Birds$ is suitable for classifying birds.

To determine the appropriate connector, we designed a positive co-occurrence matrix and a negative co-occurrence matrix to explore possible feature conflicts. The co-occurrence matrix captures the occurrence of pairs of features together in a certain rule. Specifically, consider the sample-level rule set R_k^c for class c in client k, where each rule r is defined as $f_1 \wedge \neg f_2 \wedge \cdots \wedge f_n$:

1. **The Positive Co-occurrence Matrix** records the number of times features f_i and f_j appearing together in rules in the form of $f_i \wedge \cdots \wedge f_j$. If features f_i and f_j jointly appear in rules for t times, $p_{ij} = p_{ji} = t$.

$$M_{pos} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}. \tag{1}$$

2. **The Negative Co-occurrence Matrix** records the number of times features f_i and f_j appear together in rules in the form of $f_i \wedge \cdots \wedge \neg f_j$. If features f_i and $\neg f_j$ appear in rules for t times, $q_{ij} = t$. But unlike the positive co-occurrence matrix, in the negative co-occurrence matrix, $q_{ij} \neq q_{ji}$. q_{ij} records the number of times f_i and $\neg f_j$ appear in a rule, while q_{ji} records the co-occurrence of f_j and $\neg f_i$.

$$M_{neg} = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix}. \tag{2}$$

We proposed two criteria to evaluate the extent of feature conflict through the positive co-occurrence matrix M_{pos} and the negative co-occurrence matrix M_{neg}.

1. **Diagonality**, which is calculated as:

$$D = \frac{\sum_{i=1}^{n} p_{ii}}{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}} \tag{3}$$

where n is the total number of features. A high diagonality suggests that a feature is likely to appear in a rule with no other features than itself (i.e., a high likelihood of feature conflict).

2. **Exclusivity**, which is calculated as:

$$E = \frac{\max_{i=1}^{n} \left(\sum_{j=1}^{n} q_{ij} \right)}{\sum_{m=1}^{M} l_{r_m}} \tag{4}$$

where l_{r_m} is the length of rule r_m and M is the total number of rules contributing to the negative co-occurrence matrix M_{neg}. The exclusivity E measures the average feature negative co-occurrence in a client. A high exclusivity indicates that a rule is likely to take the form of $f_i \wedge \neg f_j \wedge \cdots \wedge \neg f_n$, indicating potential feature conflicts.

When either diagonality or exclusivity exceeds a predefined hyperparameter threshold , the connector is designated as OR due to the heightened probability of feature conflicts. Otherwise, it is determined as AND. Both diagonality and exclusivity are computed on the client side based on their sample-based rules. During global rule aggregation, the FL server employs a majority voting mechanism from all client logical connectors to determine the optimal global connector. Note that, in most scenarios, the adopted logical connectors are the same for all FL clients. This consensus arises since the process of determining the logical connectors hinges on potential feature conflicts, which is an intrinsic quality of the features that remains consistent regardless of data distribution. Given that all clients share the same feature space, their assessments concerning potential feature conflicts are likely to align.

Federated Rule Aggregation

In each round of FL training, the server receives local models and local logic rules from the clients. The server first determines the appropriate logical connector following the approach described in the previous section. It then identifies a subset of rules that optimises model performance. For a given training round, the FL server receives K rules pertaining to class c from K FL clients. In practice, the maximum number of distinct rules the server can obtain is fewer than K for several reasons. Firstly, not every client is capable of generating reliable rules. We establish an accuracy threshold to filter out rules (i.e., only models exceeding the given threshold are deemed reliable). Rules derived from models with sub-par accuracy are deemed as not credible. Secondly, even if a client model meets the accuracy threshold, it might contain biased data which does not support the predicted class c. Consequently, no rule is generated for class c. Lastly, it is possible for multiple clients to produce identical rules.

To identify the optimal combination of rules, we leverage the beam search algorithm (Lowerre and Reddy 1976), a greedy approach commonly adopted in natural language processing and machine translation. Instead of exploring every possible sequence, beam search maintains the top t sequences at each step and extends them further. This approach offers a trade-off between computational cost and solution quality. In `LR-XFL`, the sequences of rules are ranked based on the rule accuracy values with respect to the validation dataset on the FL server.

FL Client Weight Assignment

In FL, assigning appropriate weights to clients is pivotal, especially when dealing with biased or noisy datasets. Under `LR-XFL`, we introduce a novel method to compute client weights predicated on the logic rules from each client. The weight assigned to a client for a given training round is set to be directly proportional to the frequency with which its rules are selected by the server. Suppose in one round, client k constructs C rules across C classes (some rules might be empty). Out of these rules, p rules are aggregated into the global rule set. The weight for client k in this round is:

$$w_k = \frac{p_k}{\sum_{i=1}^{K} p_i}. \tag{5}$$

If a client fails to generate any valid rules, or if its rules are not selected by the server, w_k is set to 0. There are two reasons for a rule to be excluded from the global set: 1) the low accuracy of the rule, and 2) its integration might compromise the effectiveness of the current global rules. A high value of w_k indicates that the client has made significant contributions in terms of rules across multiple classes in this training round, and thus shall be assigned a higher weight.

Time Complexity Analysis

In the rule generation process for K clients, each with N data points, the complexity for a single client to generate rules is $O(N)$. Considering C classes, the local rule aggregation comprises rule ranking with a complexity of $O(N \log N)$ and iterative inclusion with a complexity of $O(N)$. Therefore, the overall complexity for local rule aggregation is $O(N \log N)$. During the global rule aggregation phase, beam search is employed with a beam width of b, leading to a worst-case complexity of $O(bK^2)$ for each class. However, since N is significantly larger than K, the time complexity for the entire rule generation and aggregation process is $O(N \log N)$.

Experimental Evaluation

To evaluate the effectiveness of `LR-XFL`, we conduct experiments on four distinct datasets comparing it against three alternative approaches. Additionally, we assess `LR-XFL`'s performance under noisy data conditions. The outcomes are compared using three metrics: 1) model accuracy, 2) rule accuracy, and 3) rule fidelity. An ablation study is also conducted to highlight the importance of the logical connector selection method in `LR-XFL`.

Experiment Settings

Following the dataset settings (Barbiero et al. 2022), we adopt four benchmark datasets: 1) MNIST(Even/Odd) (LeCun 1998), 2) CUB (Wah et al. 2011), 3) V-Dem (Coppedge et al. 2022) and 4) MIMIC-II (Saeed et al. 2011). MNIST and CUB are designed following the "image $\rightarrow$ features $\rightarrow$ classes" setting. V-Dem and MIMIC are tabular datasets that directly map input to classes. To clarify, in MNIST(Even/Odd), the dataset is augmented from the original "image $\rightarrow$ digits" pattern in MNIST into the "image $\rightarrow$ digits $\rightarrow$ parity" pattern, where the digits are extracted features and the parity is the final prediction. For each of the datasets, we create two different data settings: 1) a centralised setting, serving as a baseline representing non-federated learning; and 2) a federated data setting (McMahan et al. 2017), which ensures a uniform distribution of data across FL clients. Using the MNIST dataset as an illustrative example: in the federated data setting, each of the 10 clients holds a random 10% of the entire dataset.

In order to emulate real-world scenarios, we introduce noises into the client datasets. We choose $t\%$ of clients from the entire pool of K clients, and then substitute a certain percentage of their local data with noisy data created through random label shuffling. The noise level $t\%$ is incrementally raised from 20% to 80% in the increment of 20%. We then evaluate the performance of both the global model and global rules using the server's test dataset.

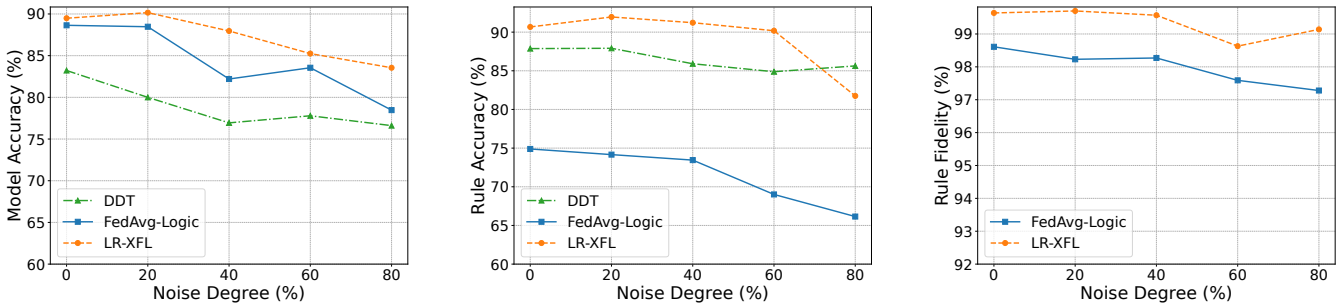
Comparison Baselines

Since there is no existing work that is specifically designed to generate and aggregate rules about FL local models and leverage such rules for FL model aggregation like `LR-XFL`, we compare `LR-XFL` against the following three baseline approaches in our experiments:

1. **Centralised Learning**: This method applies the entropy-based logic explanations of neural network (Barbiero et al. 2022) under the centralised setting with direct access to all data. It uses the OR operator to aggregate sample-level rules into class-level rules.

		Centralised Learning	Distributed Decision Tree (DDT)	FedAvg-Logic	LR−XFL
MNIST	model accuracy	99.84%	99.78%	99.80%	**99.96%**
	rule accuracy	99.84%	99.71%	99.84%	**99.95%**
	rule fidelity	99.95%	-	99.89%	**99.97%**
CUB	model accuracy	82.20%	83.22%	88.64%	**89.49%**
	rule accuracy	91.71%	87.87%	74.89%	**90.67%**
	rule fidelity	99.71%	-	98.61%	**99.64%**
V-Dem	model accuracy	93.32%	92.55%	90.95%	**93.08%**
	rule accuracy	90.84%	92.52%	86.71%	**93.08%**
	rule fidelity	94.59%	-	81.11%	**100.00%**
MIMIC-II	model accuracy	76.96%	76.40%	80.89%	**82.02%**
	rule accuracy	66.56%	67.80%	**68.27%**	65.15%
	rule fidelity	77.24%	-	**79.77%**	79.21%

Table 1: Experiment results. We mark the best performance in bold. '-' indicates that the given evaluation metric is not applicable for an approach.



(a) Model accuracy (b) Rule accuracy (c) Rule fidelity

Figure 3: Experiment results under different noise level settings.

2. **Distributed Decision Tree (DDT)** (Quinlan 1986): In the DDT approach, each client maintains its own decision tree. After local training, clients share their respective decision trees with the server. The server then evaluates the received local decision trees on a validation dataset to identify the optimal tree. This top-performing local tree is subsequently used as the global decision tree for both predictions and rule generation. As decision trees are not an iterative process, clients are restricted to using the global decision tree without any further optimisation. Decision trees naturally produce rules. In our experiments, if features (considering f_i for instance) in the rules exceed the decision tree's split value, they are represented as f_i. Conversely, features f_j in rules that are below the split value are denoted as $\neg f_j$.
3. **FedAvg-Logic**: It is an adapted version of FedAvg (McMahan et al. 2017) using the entropy-based network (Barbiero et al. 2022) as the base model. It uses the OR operator to connect for rule aggregation. Clients are assigned the same weights regardless of the performance of their rules.

Evaluation Metrics

We compare the performance of LR−XFL and the baselines using the following evaluation metrics:

1. **Classification Accuracy**: It is calculated as the number of correct predictions divided by the total number of predictions. This metric evaluates the extent to which a method accurately predicts the target outcomes.
2. **Rule Accuracy**: Rule accuracy measures the consistency between the rule predictions and the ground truth labels. Let r^c represent a rule of class c, where there are C distinct classes. Suppose the ground truth marked P data points to class c and Q data points to other classes. Among the P data points, p of them satisfy the propositions in the rule r^c, while out of the Q data points, q of them do not satisfy the propositions of rule r^c. The accuracy of this rule, denoted as $RuleAcc_c$, is defined as:
$$RuleAcc_c = \frac{p+q}{P+Q}, \tag{6}$$
The overall rule accuracy $RuleAcc$ is defined as the averaged $RuleAcc_c$ across all classes.
3. **Rule Fidelity**: Rule fidelity assesses the consistency between rule predictions and model predictions. It is computed by adapting the rule accuracy formula, where the ground truth counts P and Q are replaced with the model predicted counts of P for class c and Q for other classes.

The evaluation metrics are calculated on a test dataset stored on the server. The higher the values of these metrics, the better the performance of a given approach.

Results and Discussion

Table 1 shows the comparison results between LR−XFL and the baselines.

Model Performance Under all experiment settings, LR−XFL achieves the highest test accuracy among the federated approaches. Compared to Centralised Learning, which serves as a reference point for non-FL settings with direct access to raw data, LR−XFL often matches or even surpasses its performance. The performance can be attributed to the effectiveness of the LR−XFL automatic logical connector selection method when applied on the client side on local sample-level rules. Under FL settings, LR−XFL's advantage over FedAvg-Logic is primarily due to its client weight assignment mechanism. This mechanism effectively identifies clients whose rules make a significant contribution to the global rule set, subsequently granting them increased weights. As a result, the global server relies more heavily on these high-performing clients. Both LR−XFL and FedAvg-Logic often outperform DDT. This can be attributed to their capability to leverage insights from multiple FL clients during the training phase. In contrast, DDT selects the local model with the best rule accuracy on the validation set, without integrating local models from other FL clients. On average, LR−XFL achieves 1.19% and 3.58% higher model accuracy than FedAvg-Logic and DDT, respectively.

Rule Generation Performance Rule accuracy and rule fidelity are critical for evaluating the interpretability and trustworthiness of a model. Under FL settings, LR−XFL consistently achieves higher or comparable performance in these respects compared to FedAvg-Logic and DDT. Large differences in rule accuracy between LR−XFL and FedAvg-Logic have been observed under the CUB and V-Dem datasets. This highlights the advantage of the automatic logical connector selection method of LR−XFL. Unlike FedAvg-Logic, which aggregates local rules with the OR logical connector, LR−XFL determines the most appropriate logical connector for any given scenario. In addition, the performance gain of LR−XFL over FedAvg-Logic can also be attributed to its rule selection and the client weight assignment mechanisms during FL model aggregation. DDT, given its inherent structure that precludes rule aggregation, occasionally outperforms FedAvg-Logic by avoiding potential issues of indiscriminate rule expansion. On average, LR−XFL achieves 5.81% and 0.27% higher rule accuracy than FedAvg-Logic and DDT, respectively.

The high rule fidelity achieved by LR−XFL highlights the consistency between the rules and the FL model predictions. The rule fidelity is not applicable for DDT since it is inherently rule-based. This metric, which evaluates the alignment between rule-based and model-based predictions, is always 100% for models where rules are direct representations of the model predictions. On average, LR−XFL achieves 5.41% higher rule fidelity than FedAvg-Logic.

Resistance to Noise We also conduct experiments to study the robustness of DDT, FedAvg-Logic and LR−XFL against noise in FL clients' local data, taking the CUB dataset as a benchmark. Figure 3 illustrates the experiment results under different noise level settings. It can be observed that as the noise level increases, both model accuracy and rule accuracy decrease for all approaches. However, rule fidelity remains relatively stable, suggesting that the alignment of rules with model predictions is upheld. LR−XFL consistently achieves the highest model accuracy under various noise levels. Moreover, the degradation in its performance is moderate compared to FedAvg-Logic and DDT as the noise levels increase. Notably, the rule accuracy of LR−XFL remains relatively stable up to a noise level of 60%. This can be attributed to its rule selection and client weight assignment mechanism, which guards the global FL model against integrating information from noisy clients. Interestingly, the DDT model achieves the best performance when the noise level reaches 80%. This can be attributed to its strategy of only adopting the best single client model as the global decision tree, maximally mitigating the exposure to noise at the expense of limiting knowledge sharing among FL clients.

	LR−XFL (ablated)	LR−XFL
Model Accuracy	87.96%	**89.49%**
Rule Accuracy	76.29%	**90.67%**
Rule Fidelity	98.83%	**99.64%**

Table 2: Ablation study results.

Ablation Study

We conduct ablation studies to evaluate the impact of the automatic logical connector selection method on the performance of LR−XFL. In the experiments, threshold values for diagonality and exclusivity were set to 0.9 and 0.8, respectively, based on hyperparameter tuning. Since the logical connector chosen by LR−XFL for the CUB dataset is AND, we created an ablated version of LR−XFL using OR as the logical connector for rules under the CUB dataset. The results are shown in Table 2. It can be observed that the ablated version of LR−XFL achieves a lower model accuracy and rule fidelity, and a pronounced decline in rule accuracy compared to the full version of LR−XFL. The results demonstrate that using OR to connect rules can erroneously reduce global rule accuracy when rules may contain complementary information. This underscores the importance of the proposed automatic logical connector selection method.

Conclusions

In this paper, we proposed LR−XFL, a first-of-its-kind logic-based explainable federated learning framework. It is capable of deriving accurate global rules from local rules without requiring access to clients' local data. The most appropriate logical connectors for aggregating client rules are automatically determined by LR−XFL based on the characteristics of clients' local data. This novel design significantly enhances the trustworthiness of the resulting model. Moreover, the aggregated rules play a pivotal role in determining client weights during FL model aggregation. This transparent design enables domain experts to engage actively in the validation, refinement and adjustment of the rules, thereby helping improve the result FL model as well.

Acknowledgements

This research/project is supported, in part, by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019); the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore; and the Joint NTU-WeBank Research Centre on FinTech, Nanyang Technological University, Singapore.

References

An, Z.; and Ma, M. 2023. Guiding Federated Learning with Inferenced Formal Logic Properties. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 274–275.

Barbiero, P.; Ciravegna, G.; Giannini, F.; Lió, P.; Gori, M.; and Melacci, S. 2022. Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6046–6054.

Cha, N.; Du, Z.; Wu, C.; Yoshinaga, T.; Zhong, L.; Ma, J.; Liu, F.; and Ji, Y. 2022. Fuzzy logic based client selection for federated learning in vehicular networks. *IEEE Open Journal of the Computer Society*, 3: 39–50.

Chen, Z.; Bei, Y.; and Rudin, C. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12): 772–782.

Ciravegna, G.; Giannini, F.; Gori, M.; Maggini, M.; and Melacci, S. 2020. Human-Driven FOL Explanations of Deep Learning. In *IJCAI*, 2234–2240.

Coppedge, M.; Gerring, J.; Knutsen, C. H.; Lindberg, S. I.; Teorell, J.; Alizada, N.; Altman, D.; Bernhard, M.; Cornell, A.; Fish, M. S.; Gastaldi, L.; et al. 2022. V-Dem Country-Year Dataset v12.

Gunning, D.; Stefik, M.; Choi, J.; Miller, T.; Stumpf, S.; and Yang, G. 2019. XAI—Explainable artificial intelligence. *Science robotics*, 4(37): eaay7120.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jain, R.; Ciravegna, G.; Barbiero, P.; Giannini, F.; Buffelli, D.; and Lio, P. 2022. Extending Logic Explained Networks to Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 8838–8857. Association for Computational Linguistics.

Kairouz, P.; McMahan, H. B.; et al. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, 14(1-2): 1–210.

Kazhdan, D.; Dimanov, B.; Jamnik, M.; Liò, P.; and Weller, A. 2020. Now you see me (CME): concept-based model extraction. *arXiv preprint arXiv:2010.13233*.

Kim, B.; Gilmer, J.; Wattenberg, M.; and Viégas, F. 2018. Tcav: Relative concept importance testing with linear concept activation vectors.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept bottleneck models. In *International conference on machine learning*, 5338–5348. PMLR.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Lee, S.; Wang, X.; Han, S.; Yi, X.; Xie, X.; and Cha, M. 2022. Self-explaining deep models with logic rule reasoning. *Advances in Neural Information Processing Systems*, 35: 3203–3216.

Lowerre, B.; and Reddy, R. 1976. The harpy speech recognition system: performance with large vocabularies. *The Journal of the Acoustical Society of America*, 60(S1): S10–S11.

Lundberg, S. M.; and Lee, S. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

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

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning*, 1: 81–106.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " 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*, 1135–1144.

Saeed, M.; Villarroel, M.; Reisner, A. T.; Clifford, G.; Lehman, L.; Moody, G.; Heldt, T.; Kyaw, T. H.; Moody, B.; and Mark, R. G. 2011. Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database. *Critical care medicine*, 39(5): 952.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Stammer, W.; Schramowski, P.; and Kersting, K. 2021. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3619–3629.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Xu, F.; Uszkoreit, H.; Du, Y.; Fan, W.; Zhao, D.; and Zhu, J. 2019. Explainable AI: A brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*, 563–574. Springer.

Yu, H.; Miao, C.; An, B.; Shen, Z.; and Leung, C. 2014. Reputation-aware Task Allocation for Human Trustees. In *Proceedings of 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14)*, 357–364.

Zhu, X.; Wang, D.; Pedrycz, W.; and Li, Z. 2021. Horizontal federated learning of Takagi–Sugeno fuzzy rule-based models. *IEEE Transactions on Fuzzy Systems*, 30(9): 3537–3547.