Federated Inductive Logic Programming for Explainable Artificial Intelligence

Yasmine Akaichi, Jean-Marie Jacquet, Isabelle Linden and Wim Vanhoof*

Faculty of Computer Science, University of Namur, Belgium

Abstract

Artificial Intelligence (AI) is transforming healthcare through predictive modeling and clinical decision support. However, its adoption remains limited due to two persistent challenges: the lack of model explainability and the sensitivity of patient data, which restricts data sharing across institutions. Federated Learning (FL) offers a privacy-preserving solution by enabling collaborative model training across decentralized health data sources without exposing raw data. Yet, most FL implementations rely on black-box models, such as deep neural networks, which limits clinical transparency. While interpretable models like Decision Trees (DT) exist, they often lack the expressiveness required to capture complex medical relationships. To bridge this gap, we propose FILP: a novel framework that integrates Explainable AI (XAI) via Inductive Logic Programming (ILP) into a Federated Learning setting. FILP is applied to a real-world clinical task—predicting post-intubation complications. It builds on the symbolic and declarative nature of logic programming, to derive logical rules from local data. Each client independently learns interpretable theories using background knowledge and examples, then participates in a majority-vote consensus mechanism to produce a global theory that is both privacy-preserving and human-understandable. We instantiate FILP using two ILP systems: Andante , a Progol-based system, and Popper, which is based on Answer Set Programming (ASP). FILP is evaluated against decision tree baselines in both centralized and federated settings. While decision trees achieve strong predictive results on small test sets, ILP methods offer a distinct advantage in interpretability and explainability by producing symbolic, domain-aligned rules. Our results indicate that, in this clinical setting, federated training yields slightly improved performance over centralized learning, while preserving privacy. This suggests that FL can offer not only confidentiality guarantees but also potential advantages in predictive effectiveness.

1. Introduction

Artificial Intelligence (AI) is increasingly integrated into healthcare workflows, where it supports tasks such as diagnosis, prognosis, and treatment planning [1]. Predictive models trained on electronic health records (EHRs) [2], medical imaging [3], and physiological signals [4] have shown promise in identifying risk patterns and improving outcomes.

Yet, despite these advances, real-world clinical deployment remains limited. Two core challenges persist: the lack of model explainability and the constraints imposed by patient data privacy [5, 6]. Modern healthcare institutions operate in a distributed setting, where data is siloed across hospitals. This makes centralized training impractical and creates demand for AI systems that are both privacy-preserving and clinically interpretable.

Federated Learning (FL) addresses data sharing challenges by enabling decentralized model training without exposing raw data [6, 7]. FL has shown promise in healthcare applications, including medical imaging and EHR analysis [8, 9], but most implementations rely on black-box models such as deep neural networks [10], limiting transparency and clinical validation. In response, interpretable models like decision trees have been proposed as white-box alternatives [11]. While decision trees provide structural interpretability by mapping predictions to feature-based splits, they fall short in modeling rich

EXPLIMED 2025 - Second Workshop on Explainable Artificial Intelligence for the Medical Domain - 25–30 October 2025, Bologna, Italy

*Corresponding author.

yasmine.akaichi@unamur.be (Y. Akaichi); jean-marie.jacquet@unmaur.be (J. Jacquet); isabelle.linden@unmaur.be (I. Linden); wim.vanhoof@unmaur.be (W. Vanhoof)

https://www.unamur.be/en/profil/yakaichi (Y. Akaichi); https://unamur.be/fr/profil/jacquejm (J. Jacquet); https://www.unamur.be/en/profil/ilinden (I. Linden); https://www.unamur.be/fr/profil/wvanhoof (W. Vanhoof)

0009-0005-5078-1583 (Y. Akaichi); 0000-0001-9531-0519 (J. Jacquet); 0000-0001-8034-1857 (I. Linden)

relational dependencies and fail to incorporate structured domain knowledge in a reusable, declarative form.

This introduces a methodological gap: FL systems are often optimized for accuracy, but lack support for semantically grounded, expert-verifiable reasoning [12, 13]. In high-stakes domains like healthcare, this trade-off is critical.

Symbolic AI, particularly Inductive Logic Programming (ILP), offers a complementary solution. ILP induces logic programs from labeled examples and background knowledge, producing hypotheses in the form of interpretable rules [14, 15]. Unlike decision trees, ILP supports structured, relational learning and natively integrates prior knowledge into the learning process. ILP has demonstrated utility in biomedical domains such as toxicology [16], pharmacology [17], and genomics [18]. However, existing ILP systems typically assume centralized access to data and are not designed to handle the decentralization, heterogeneity, and coordination challenges of federated environments.

In this paper, we propose **FILP**, a novel framework that integrates ILP into Federated Learning to enable symbolic, interpretable modeling from decentralized medical data. Each client learns logical rules locally, guided by background knowledge and declarative biases, and shares only symbolic predictions on its data. A majority-vote aggregation mechanism is then used to derive a global consensus decision. FILP is privacy-preserving by design and enables human-understandable model inspection, a key factor in clinical adoption. Our main contributions are as follows:

- We introduce FILP, the first framework to bring ILP into the federated setting, addressing privacy and regulatory constraints (e.g., GDPR, AI Act) while preserving interpretability.
- We implement FILP with two ILP engines, Andante (Progol-based) and Popper (ASP-based), demonstrating generality across distinct symbolic reasoning paradigms.
- We evaluate FILP on a real-world clinical dataset, showing that it induces clinically plausible rules even under strict privacy and limited data availability.
- We extend the evaluation to three established ILP benchmarks (Mutagenesis, Carcinogenesis, Alzheimer), highlighting FILP's ability to generalize beyond a single domain.

This paper is structured as follows. Section 2 introduces the background on Federated Learning and ILP. Section 3 presents the FILP framework and its components. Section 4 reports experimental results on both clinical and benchmark datasets. Finally, Section 5 concludes with a summary of contributions and future research directions.

2. Background

In this section, we review the key concepts and formal principles that support our work. We begin with an overview of FL, followed by a discussion of explainable FL in healthcare. We then introduce ILP, a symbolic machine learning paradigm for interpretable model learning.

2.1. Federated Learning

Federated Learning is a decentralized machine learning paradigm introduced by McMahan et al. [10, 7], designed to enable collaborative model training across multiple clients without sharing raw data, thereby preserving privacy. Each client trains a model locally on its private dataset and sends updates to a central server, which aggregates them to produce a global model that benefits from the collective knowledge of all participants.

FL Components. A typical FL setup includes three important components: parties or clients (e.g., mobile devices or institutions), the manager or the coordinator (e.g., central server), and the communication-computation framework responsible for training the machine learning models [19].

FL Training. Federated training proceeds in synchronous communication rounds. At the beginning of round t, the server S samples a subset of clients $S_t \subseteq \{C_1, \ldots, C_n\}$ and broadcasts the current global parameters θ_t. Each selected client C_k trains locally on its private data for a fixed number of epochs and obtains updated parameters θ_{t+1}^k. The server then aggregates the client models using Federated Averaging (FedAvg):

$$m_t = \sum_{k \in S_t} n_k, \qquad \theta_{t+1} = \sum_{k \in S_t} \frac{n_k}{m_t} \theta_{t+1}^k,$$

where n_k is the number of local training examples on client C_k. This weighted average ensures that clients with larger datasets contribute proportionally to the global update. The procedure repeats for multiple rounds until convergence.

FL naturally supports partial participation, unbalanced sample sizes, and non-IID data. However, standard FL assumes differentiable, numeric model parameters and gradient-based optimization. In contrast, ILP produces discrete symbolic hypotheses (logic programs), which cannot be averaged as above. This incompatibility motivates our symbolic FL framework (FILP), where we exchange and aggregate *hypotheses* rather than gradient-based parameter updates.

2.2. FL for XAI

FL's applications in healthcare span a broad range of domains [6], including electronic health records (EHRs), such as arrhythmia detection from single-lead ECG signals collected via IoT devices [20], medical imaging [21], COVID-19 diagnosis from multimodal data [22], and smart health systems for early Alzheimer's detection [23].

Recent studies tend to focus on either interpretability or privacy preservation, but rarely address both simultaneously. Many approaches rely on post-hoc explainability techniques such as SHAP [24] or LIME [5, 25], which attempt to rationalize predictions after model training. For instance, Janzing et al. [26] integrated gradient-based explanations to help users understand whether a specific feature positively or negatively influences a decision. Similarly, Xu et al. [27] applied feature selection for interpretable load forecasting. Zeleke and Bochicchio [28] integrate explainable AI into FL for arrhythmia detection by using an attention-based temporal convolutional network (TCN), where attention weights are extracted post-training to highlight which parts of the ECG signal influenced the model's predictions.

While methods like SHAP and LIME attempt to explain black-box predictions through feature attribution, they remain external to the learning process and rely on approximations. This lack of integration, coupled with susceptibility to adversarial manipulation, raises concerns in high-stakes domains like healthcare. As Rudin [29] argues, inherently interpretable models should be preferred over post-hoc explanations, as they offer faithful and verifiable insights directly grounded in model structure.

To address these limitations, some recent efforts have introduced symbolic components into FL. FedNSL [30] and symbolic regression-based methods [31] generate symbolic expressions, but still rely on neural approximations. PeFLL [32] enhances personalization through client embeddings, but retains a black-box structure. These approaches offer partial improvements but fall short of full interpretability and logical rigor. A more principled solution lies in the use of inherently interpretable models.

Decision trees, for example, have been incorporated into FL frameworks [33, 34] to promote transparency. Wu et al. [35] introduced FedForest, a federated random forest algorithm designed to balance predictive accuracy and interpretability. While decision trees can express simple conjunctive rules over tabular features, such as "if a patient is diabetic and has a short neck, then complications are likely," they remain limited in their ability to capture richer relational structures or integrate background domain knowledge. Decision trees operate through isolated feature splits and lack the representational flexibility of logic-based systems, making it difficult to model reusable abstractions, relationships between entities, or derived clinical concepts in a declarative way.

Of course, rule-based modeling in healthcare encompasses a broader spectrum of approaches than those discussed here. In this section, however, we focus on representative families that are both widely

applied and closely aligned with the challenges of federated learning and explainability.

Thus, another widely explored class of rule-based systems in healthcare is fuzzy logic. These models are among the most established approaches to building interpretable systems, as they capture knowledge in the form of human-readable IF–THEN rules with linguistic terms (e.g., low, medium, high). Their ability to handle uncertainty and imprecision has made them particularly attractive in healthcare and other sensitive domains where transparency is essential. Works such as [36] adopt fuzzy logic in a centralized way, where interpretability is achieved directly through fuzzy rules. By contrast, Castellano et al. [37] focus on enhancing the interpretability of deep learning models by integrating fuzzy logic with graph neural architectures. Their approach first fuzzifies input features into linguistic terms, constructs graph representations of the data, and then applies a GNN for classification.

Other work [38, 39, 40, 41] has adopted fuzzy rule-based systems in a federated learning context. In particular, Ducange et al. [42] focused on Takagi–Sugeno–Kang Fuzzy Rule-Based Systems (TSK-FRBS), which naturally provide interpretability through fuzzy IF–THEN rules. They contrasted this approach with a multi-layer perceptron neural network (MLP-NN), where explainability was obtained only through post-hoc methods. In their framework, local models trained at different clients were aggregated on the server by merging rule bases and resolving conflicts via a weighted averaging mechanism that accounts for rule support and confidence. Their experiments on Alzheimer's disease prediction highlighted the benefits of interpretable fuzzy rules in FL.

More recently, Barcena et al. [43] proposed a framework for FL of Fuzzy Regression Trees (FRTs), aiming to jointly address privacy preservation and model interpretability. Their method builds a single global tree by aggregating local statistics (e.g., fuzzy means, variances, and membership degrees) instead of raw data, thereby enabling the construction of an interpretable tree structure across distributed clients.

These contributions demonstrate that fuzzy logic has been successfully applied in both centralized and federated learning, offering interpretable models through fuzzy rules or tree structures. By contrast, our work introduces ILP into FL, where interpretability arises directly from first-order logical rules rather than fuzzy sets. Moreover, instead of relying on statistical or weight-based aggregation, we adopt a majority voting mechanism across distributed hypotheses, emphasizing consensus-driven model construction. This positions our contribution as a complementary path toward explainable FL, centered on symbolic logic expressiveness and federated consensus.

2.3. ILP for XAI

In the XAI, it is useful to discern between *interpretability* and *explainability* [44]. Interpretability refers to the fact that the structure of a model can be directly understood by humans (e.g., reading a set of rules). Explainability goes one step further: it requires understanding *why* a specific prediction was made. ILP naturally supports both dimensions. It induces logic rules that are declarative, human-readable, and grounded in domain knowledge [45, 46, 14].

Unlike data-intensive statistical models, ILP can generalize from relatively few examples. Its output is a set of rules that align with expert reasoning and can be verified logically. This makes ILP particularly attractive for XAI, since it provides transparency, supports domain knowledge integration, and enables transfer or lifelong learning [47, 48].

ILP has been applied in several explainability contexts, for example to extract interpretable rules from black-box models, explain CNN decisions [49, 50], and uncover biases in real-world tasks. Despite its strengths, ILP faces challenges with scalability, noise tolerance, and efficient hypothesis search [51]. Recent advances combine ILP with statistical relational learning and neural-symbolic methods to improve robustness while retaining interpretability. ILP's versatility extends across domains: in legal reasoning [52], reinforcement learning [53] and fraud detection [54]. In healthcare, ILP has produced interpretable models for brain tumor classification [55], and toxicology [18].

However, classical ILP systems assume centralized data and lack mechanisms for decentralized training. Second, they struggle with scalability, as hypothesis search quickly becomes computationally expensive when the amount of data or background knowledge grows [51]. Finally, symbolic rules cannot

be optimized with gradient descent, which makes ILP difficult to integrate into standard federated learning architectures. These limitations motivate our proposed framework, FILP, which integrates ILP into a federated setting to enable decentralized, explainable, and scalable learning.

2.4. ILP Systems

Our proposed system FILP is designed to work with different ILP systems. In particular, it can integrate either **Progol**, which induces rules by generalizing from examples, or **Popper**, which searches the hypothesis space under constraints.

ILP Problem Setting ILP combines logic programming and machine learning to derive interpretable rules from structured data. Given background knowledge B_k, a set of positive examples E^+, and a set of negative examples E^-, the task is to induce a hypothesis H (a logic program) such that:

$$\forall e^+ \in E^+ \,:\, B_k \cup H \models e^+ \quad \text{and} \quad \forall e^- \in E^- \,:\, B_k \cup H \not\models e^-.$$

In other words, the induced hypothesis should entail all positive examples while excluding all negative ones.

2.4.1. Progol.

Progol [56] is based on *inverse entailment*. For each positive example $e \in E^+$, it constructs the most specific clause that explains the example relative to the background knowledge (called the *bottom clause*$\perp$). It then generalizes this clause into a rule that (i) covers the example, (ii) avoids all negative examples, and (iii) respects the syntactic bias defined by mode declarations and determinations.

Algorithm 1 summarizes the learning procedure. The algorithm iteratively processes positive examples, generates a bottom clause, generalizes it into a rule consistent with the negative examples, and updates the hypothesis accordingly.

Algorithm 1: Progol [56]

Input: B_k: background knowledge, including mode declarations
E^+: set of positive examples
E^-: set of negative examples
Output: H, a hypothesis consisting of a set of clauses
S := E^+
H := $\emptyset$
while $S \neq \emptyset$ **do**
  select example e from S
  construct bottom clause $\perp$ from B_k and e
  induce clause h generalizing $\perp$ and consistent with E^-
  $H := H \cup \{h\}$
  remove all $x \in S$ such that $(B_k \cup H) \models x$

Progol has been widely applied in relational learning tasks such as biomedical reasoning, natural language processing [57], and molecular property prediction [58]. Its output is a set of symbolic rules that are both human-readable and logically verifiable.

2.4.2. Popper.

Popper [46] approaches ILP as a constraint-guided search problem. Instead of starting from specific examples, Popper incrementally explores the space of possible hypotheses under syntactic constraints such as clause length. Each candidate program is tested against positive and negative examples. When a candidate fails, new constraints are learned to prune similar programs from the

search space. This iterative process continues until a valid program is found or the search space is exhausted. Algorithm 2 summarizes the procedure. The algorithm increases clause length step by step, generates candidate programs, tests them against examples, and refines the constraints accordingly. !

Algorithm 2: Popper [?]

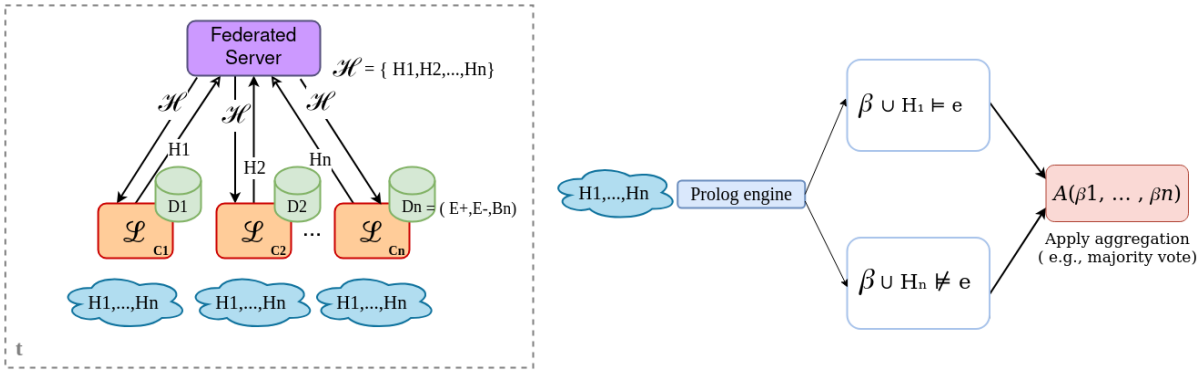
```
def popper(ℬ, ℰ, declaration_bias, constraints, max_literals):
    num_literals ← 1
    while num_literals ≤ max_literals do
        program ← generate(declaration_bias, constraints, num_literals)
        if program == 'space_exhausted' then
            num_literals += 1
            continue
        outcome ← test(ℬ, ℰ+, ℰ−, program)
        if outcome == ('all_positive', 'none_negative') then
            return program
        constraints += learn_constraints(program, outcome)
    return {}
```

Popper is particularly suited for tasks that require compact, minimal programs with explicit control over the search space. It has been used in program synthesis, strategy learning [59], and relational reasoning over structured symbolic domains [?].

In contrast to Progol's example-driven strategy, Popper systematically enumerates and prunes hypotheses using constraints. FILP's compatibility with both paradigms illustrates its generality and its capacity to support multiple ILP workflows in federated symbolic learning.

3. Proposed Methodology

We propose a Federated Inductive Logic Programming (FILP) framework that addresses both data privacy and explainability in a decentralized learning setting. FILP enables multiple clients to collaboratively induce symbolic rules without sharing raw data, while maintaining interpretability across both the training and inference phases. To the best of our knowledge, this is the first FL approach that aggregates ILP hypotheses without requiring numerical model updates.



Part (a) : Federated ILP Learning

Part (b) : Prediction

Figure 1: FILP architecture: **Part(a)** During training, each client independently induces a local hypothesis (set of symbolic rules) from its examples and background knowledge using an ILP system (Andante or Popper). These hypotheses are shared with the server, which redistributes the ensemble back to all clients without raw data exchange. **Part(b)** During prediction, each client evaluates incoming examples against the distributed hypotheses using its background knowledge, and applies a logic-based aggregation function (e.g., majority vote) to derive the final decision.

3.1. Problem Setting

We consider a scenario with a set of n clients $\mathscr{C} = \{C_1, C_2, \ldots, C_n\}$. Each client C_i holds a private local dataset $\mathscr{D}_i = \langle E_i^+, E_i^-, \mathscr{B}_i \rangle$, where E_i^+ and E_i^- are the sets of positive and negative examples respectively, and $\mathscr{B}_i$ is the background knowledge expressed in logic programming form. Background knowledge remains local and is not shared across clients. The datasets share a common schema but remain decentralized. A central server S coordinates the process without accessing any raw data. Figure 1 summarizes the FILP framework, which operates in two phases: *(a)* symbolic hypothesis induction at the client side, and *(b)* decentralized prediction via logic-based aggregation. This setup is formalized through three main components:

- $\mathscr{L}$ — the ILP learning engine (e.g., Popper or Andante) that returns a hypothesis (a set of clauses/rules).
- A — the aggregation function (e.g., majority vote over hypotheses on the client side).
- t — an FL framework ensuring the collaboration. (e.g., flower).

Combining ILP with FL introduces unique challenges. Symbolic models, such as those induced by ILP, are discrete, heterogeneous, and inherently non-differentiable, meaning they cannot be optimized through gradient descent or averaged as in traditional neural network-based FL. This non-differentiability prevents the use of standard FL aggregation techniques such as federated averaging. Instead, logic programs require specialized aggregation strategies that respect their syntactic structure and semantic meaning. To the best of our knowledge, no standard approach exists for aggregating logic programs in federated settings.

3.2. FILP Overview

To address these limitations, FILP combines ILP with a distributed, logic-based voting mechanism. As shown in Algorithm 3, each client independently induces a set of symbolic rules (Hypothesis, also called Theory in some work) using an ILP learner (e.g., Andante or Popper) over its local examples and background knowledge. These local hypotheses, expressed as logic programs, are transmitted to a central server, which aggregates them not by parameter averaging but by redistributing the full set of candidate hypotheses to all clients. Rather than fusing rules into a single theory, which may introduce contradictions, we redistribute the ensemble to preserve local semantic consistency and allow flexible voting.

During the prediction phase shown in algorithm 4, each client evaluates whether each hypothesis in the received ensemble entails the query example under its own background knowledge. The predictions are then combined via a majority vote, producing a federated decision that preserves privacy, supports symbolic reasoning, and yields interpretable outputs.

3.2.1. Part (a): Federated ILP Training

At the beginning of the learning phase, the server S broadcasts global ILP settings, such as mode declarations and type constraints, to all clients. Each client C_i then uses its local dataset $\mathscr{D}_i = \langle E_i^+, E_i^-, \mathscr{B}_i \rangle$ to induce a symbolic hypothesis H_i using the ILP learner, i.e., $H_i \leftarrow \mathscr{L}(E_i^+, E_i^-, \mathscr{B}_i)$. In our framework, the ILP engine $\mathscr{L}$ can be instantiated with either Andante or Popper, two well-known systems described in Section 2.4. Each hypothesis H_i consists of logic rules such as:

$H_i = \{$ `complication(X) :- hta(X), diabetes(X).`, `complication(X) :- age(X,A), A>65.` $\}$.

These hypotheses are sent to the server, which aggregates them into an ensemble $\mathscr{H} = \{H_1, \ldots, H_n\}$ and redistributes this set to all clients for inference. Since FILP performs only a single round of federated training, there is no iterative refinement or communication of gradients, making the process lightweight and communication-efficient.

Algorithm 3: The FILP($\mathscr{L}$) learning algorithm

Input: ILP system $\mathscr{L}$ (e.g., Andante or Popper)
Server S and clients $\{C_1, \ldots, C_n\}$
Global ILP settings (mode declarations, type constraints); client $\mathscr{B}_i$ remains local
Datasets $\mathscr{D}_1, \ldots, \mathscr{D}_n$, where $\mathscr{D}_i = \langle E_i^+, E_i^-, \mathscr{B}_i \rangle$
Output: An ensemble of logic theories $\mathscr{H} = \{H_1, \ldots, H_n\}$

Server executes:
Broadcast ILP settings to each client C_i

Each Client C_i executes:
Use $\mathscr{L}$ to learn hypothesis $H_i \leftarrow \mathscr{L}(E_i^+, E_i^-, \mathscr{B}_i)$
Send H_i to the server

Server executes:
Collect all H_i from clients
Aggregate into hypothesis ensemble $\mathscr{H} = (H_1, \cdots, H_n)$
Send $\mathscr{H}$ to all clients

3.2.2. Part (b): Prediction via Aggregated Hypotheses

Each client performs a local prediction using the received hypothesis ensemble $\mathscr{H}$, its background knowledge $\mathscr{B}_i$, and a local test example e. The actual logical inference is delegated to an internal reasoning module (e.g., a local Prolog engine), which remains fully local and does not require external communication.

For each hypothesis $H_i \in \mathscr{H}$, the client queries whether $H_i \cup \mathscr{B}_i \models e$, returning $\beta_i \in \{\texttt{true}, \texttt{false}\}$. This entailment is checked via standard logical inference:

$$\beta_i = \texttt{PrologQuery}(H_i \cup \mathscr{B}_i, e).$$

This corresponds to checking whether $H_i \cup \mathscr{B}_i \models e$. The binary outcome $\beta_i \in \{\texttt{true}, \texttt{false}\}$ reflects whether the example is entailed by that specific hypothesis under the client's background.

Both ILP systems supported in FILP (Andante and Popper) rely on a Prolog-based runtime to evaluate entailment. While their learning strategies differ, their prediction phase shares this symbolic backbone, ensuring consistency and traceability.

Algorithm 4: FILP-Predict($\mathscr{H}, \mathscr{B}_i, e$)

Input: Hypothesis set $\mathscr{H} = \{H_1, \ldots, H_n\}$
Local background knowledge $\mathscr{B}_i$
Test example e
Output: Prediction decision for example e

Initialize list of entailment results $\boldsymbol{\beta} \leftarrow [\,]$
foreach $H_i \in \mathscr{H}$ **do**
  `// Query if the hypothesis entails the example`
  $\beta_i \leftarrow \texttt{PrologQuery}(H_i \cup \mathscr{B}_i, e)$
  Append β_i to $\boldsymbol{\beta}$; `// true or false`

`// Aggregate individual entailment decisions`
Return $A(\beta_1, \ldots, \beta_n)$; `// e.g., majority vote`

Local Reasoning and Privacy This reasoning is executed locally on each client, without transmitting test data or background knowledge. This ensures that privacy is preserved, as only the symbolic hypotheses are exchanged in the FL process.

Aggregation via Majority Vote After evaluating all hypotheses, each client applies an aggregation function (e.g., majority vote) to produce a final prediction:

$$A(\beta_1, \ldots, \beta_n) = \begin{cases} \texttt{true}, & \text{if the majority of hypotheses in } \mathscr{H} \text{ entail } e, \\ \texttt{false}, & \text{otherwise.} \end{cases}$$

Although majority voting is used in this work, other aggregation functions (e.g., weighted voting, rule selection) are possible and left for future investigation.

Interpretability at Prediction Time This strategy allows clients to collaboratively reach a decision based on diverse symbolic perspectives, while retaining traceability. FILP provides interpretability at two levels:

- **Model-level**: the learned hypotheses consist of rule-based logic programs that are human-readable and can be directly interpreted by clinicians or domain experts.
- **Prediction-level**: Although each hypothesis H_i consists of interpretable rules, aggregating their predictions via majority vote can obscure the specific reasoning behind a classification decision. To mitigate this, our system supports fine-grained traceability during inference:
  - For each test example, clients log which rules from each H_i entail the target predicate. This enables local explainability, where domain experts can audit the exact rules that were used.
  - We display per-rule coverage and fidelity statistics alongside each example, helping assess both the predictive and explanatory value of each rule. We also report explainability metrics such as rule length, number of clauses (or rules), and coverage, as detailed in Section 4.

By replacing opaque prediction with declarative reasoning, FILP ensures that both the global model behavior and individual predictions are transparent, explainable, and interpretable.

4. Experimental Results

We empirically evaluate FILP on one real-world clinical dataset and three standard ILP benchmarks. The evaluation covers both predictive performance and explainability metrics (rule coverage, fidelity, average rule length, and number of rules), and includes comparisons against a classical machine learning baseline.

4.1. Datasets and Representations

We consider four binary classification datasets: Post-Intubation Complications, Mutagenesis, Carcinogenesis, and Alzheimer Drugs, summarized in Table 1.

Table 1
Summary of datasets (N = total number of examples, Pos = number of positives, Neg = number of negatives). Detailed partitioning across federated clients is provided in Appx. A.

Dataset	N	Pos	Neg
Post-Intubation Complications	118	72	46
Mutagenesis	230	138	92
Carcinogenesis	298	162	136
Alzheimer (Acetyl)	1060	530	530

4.1.1. Post-Intubation Complications.

The dataset consists of 118 anonymized patient records, each labeled by the presence or absence of post-intubation complications. For each patient, structured clinical features are provided, spanning anatomical factors (e.g., thyromental distance, neck morphology), physiological variables (e.g., age, obesity), comorbidities (e.g., COPD, diabetes, hypertension), and procedural aspects (e.g., intubation attempts, operator experience). Due to confidentiality restrictions, the dataset cannot be made publicly available, but it may be shared upon request under formal institutional data-sharing agreements.

4.1.2. Mutagenesis.

A relational chemistry dataset [60] where molecules are described as logical structures over atoms, bonds, and functional groups, with a binary mutagenicity label. The objective is to classify compounds according to their mutagenic activity. The target predicate is `active(Compound)`, which holds when a compound induces genetic mutations. The dataset comprises 230 nitroaromatic compounds with rich relational background knowledge enabling structure–activity reasoning.

4.1.3. Carcinogenesis.

A relational toxicology dataset [18] where compounds are represented as molecular graphs. The objective is to predict the carcinogenicity of chemical compounds in rodents. The target predicate is `active(Compound)`, which holds when a compound exhibits carcinogenic activity. The dataset comprises 298 compounds described by relational facts encoding molecular substructures, toxicophores, and physicochemical features.

4.1.4. Alzheimer Drugs.

A relational pharmacology dataset [61] where compounds are linked to pharmacological properties through drug–property relations. The objective is to identify molecules with desirable therapeutic effects for Alzheimer's treatment, particularly acetylcholinesterase inhibition. The target predicate is `great(Compound,Property)`, which holds when a compound demonstrates effective pharmacological activity. The dataset comprises 1,060 molecule–property pairs, described relationally through chemical groups and structural motifs.

4.2. Preprocessing

For the Medical (Post-Intubation Complications) dataset, preprocessing included cleaning variable names, handling missing values, and preparing two complementary representations. A tabular view was derived through categorical encoding, imputation of missing entries, and normalization of numerical variables to enable decision-tree baselines. In parallel, a logical view was generated by encoding patient attributes as Prolog predicates consistent with the background knowledge.

By contrast, the Mutagenesis, Carcinogenesis, and Alzheimer datasets are classical ILP benchmarks already provided in logical form. For these, preprocessing was limited to minor syntax adaptation between Popper and Andante, and the derivation of propositional (tabular) versions for baseline comparison.

4.2.1. Data Representations

Each dataset is prepared in two synchronized views: a Prolog/ILP view, where instances are encoded as first-order facts with background-knowledge predicates, and a tabular view, where the same information is propositionalized into a feature matrix for decision trees. The tabular view applies one-hot encoding to categorical features, imputes missing values with the most frequent value per feature (optionally standardizing numeric features), removes constant/duplicate columns, and harmonizes the feature

schema across clients. In the ILP representation, the learning task is guided by a declarative bias that constrains the hypothesis space :

Progol. For Progol (and Andante), the declarative bias is defined by `modeh`, `modeb`, and `determination` declarations. The `modeh` declaration specifies the target predicate to be learned (e.g., `complication(+patient)`), while `modeb` declarations define permissible predicates in rule bodies, and `determination` statements control dependency propagation and mode directionality.

Popper. For Popper, the bias is specified through `head_pred`, which defines the target predicate, `body_pred`, which declares the predicates admissible in rule bodies, and `direction` annotations (`in`/`out`), which constrain the flow of arguments. These syntactic constraints ensure tractability, prune the search space, and align the induced rules with domain knowledge.

These declarative biases ensure that hypotheses remain logically tractable and comparable across clients.

4.2.2. Data Partitioning

Each dataset is first split into a centralized training set (80%) and a held-out global test set (20%), using stratified sampling on the target label to preserve class balance (details in Appendix A). The centralized training set is then divided into three disjoint client partitions to simulate a cross-silo federated setting. The global test set is likewise divided into three disjoint subsets, providing a local test set for each client. Results are reported both per client on local tests and on the global test, which remains disjoint from all training data. Evaluating also on the global test allows us to assess performance on a larger, more diverse sample beyond the small client-specific tests. Centralized baselines train on the entire centralized training split and are evaluated on the same global test. Unless stated otherwise, background knowledge (BK) is kept local to each client and not shared.

4.3. Experimental Setup

We simulate a cross-silo federated learning scenario using custom orchestration framework implemented in Python. The environment consists of three virtual clients, each assigned to a distinct data partition reflecting a federated setup. The experimental pipeline includes the following components:

- **Flower [62]:** an open-source federated learning framework used to coordinate the training and aggregation of decision tree models.
- **Andante:** a custom-built ILP system based on Progol, featuring an interactive user interface that facilitates manipulation of background knowledge, examples, and induced rules.
- **Popper:** a state-of-the-art ILP system based on Answer Set Programming, which leverages a "learning from failures" strategy to explore the hypothesis space.
- **ID3 Decision Tree:** a classical decision tree learning algorithm used as a baseline model in both centralized and federated configurations.

For ILP-based experiments, each client independently learns a local hypothesis using either Andante or Popper. The resulting sets of logical rules are then evaluated on the corresponding test examples, and aggregated predictions are computed using a majority voting scheme over the local hypotheses. For the federated decision tree baseline, each client trains a local decision tree classifier, and predictions are likewise aggregated via majority voting.

All experiments adhere to fixed and consistent train/test splits across methods. ILP systems operate over structured relational representations encoded in Prolog, while decision trees are trained on conventional tabular data. Federated training and evaluation are conducted within a fully simulated local environment using our custom orchestration pipeline in Python.

4.4. Results on Post-Intubation Complications

We first present results on the Post-Intubation Complications dataset, used as a real-world clinical test case for this work. While limited in size, this dataset provides a realistic clinical setting to illustrate the behavior of FILP when applied with Popper or Andante. Despite its small scale, this case study highlights the strength of ILP methods in generalizing from very few examples while maintaining interpretability.

4.4.1. Quantitative Analysis

Descriptive Accuracy Table 2 reports descriptive accuracy, i.e., how faithfully the induced rules reproduce the logical training distribution. This metric, also referred to as fidelity, evaluates whether the hypotheses correctly entail positive training examples and reject negative ones.

The results show a consistent pattern across methods: false positives are uniformly zero, meaning neither Popper nor Andante ever misclassifies a negative example as positive. This guarantees that no false complications are predicted, and precision is therefore always 1.00.

Table 2
Coverage analysis of induced rules by ILP methods for Popper and Andante. The values of TP, FN, TN, and FP indicate how many examples are entailed or rejected by the generated hypotheses. Derived metrics such as Accuracy, Precision, and Recall thus provide descriptive insights into how well the induced rules capture the data distribution.

Method / Partition	TP	FN	TN	FP	Precision	Recall	Accuracy
ILP (Popper) – Client 1	17	2	12	0	1.00	0.89	0.94
ILP (Popper) – Client 2	15	4	12	0	1.00	0.78	0.87
ILP (Popper) – Client 3	18	1	12	0	1.00	0.94	0.97
Popper – Centralized	48	9	36	0	1.00	0.84	0.90
ILP (Andante) – Client 1	15	4	12	0	1.00	0.78	0.87
ILP (Andante) – Client 2	12	7	12	0	1.00	0.63	0.77
ILP (Andante) – Client 3	14	5	12	0	1.00	0.73	0.83
Andante – Centralized	11	46	36	0	1.00	0.19	0.50

Differences instead arise from false negatives, which directly lower recall. Popper achieves higher recall (0.89–0.94 across clients) and accuracies above 0.87, slightly outperforming its centralized counterpart. Andante performs moderately in the federated case (recall 0.63–0.78), but collapses when centralized (accuracy 0.50).

Finally, Popper consistently outperforms Andante in recall and overall accuracy. This difference is attributable to their underlying search strategies: Popper (ASP-based) employs a constraint-driven, optimal search that explores hypotheses more systematically, whereas Andante (Progol-based) relies on heuristic, greedy refinement. As a result, Popper better captures the positive class while maintaining compact rules, offering a more faithful symbolic representation of the data.

Predictive Accuracy Table 3 reports predictive accuracy for FILP with Popper and Andante, compared to decision trees. For ID3, the maximum depth was adjusted according to dataset size: for Alzheimer ($N > 500$), depth was set to 9, while for the smaller datasets ($N < 500$), depth varied between 3 and 5. For Popper, the maximum number of clauses and body literals was fixed to 6 in all experiments, ensuring a controlled hypothesis space across clients.

Results show that FILP consistently outperforms centralized ILP: Popper and Andante reach 0.60–0.75 accuracy per client, compared to 0.40–0.56 in the centralized setting. Popper exhibits more stable accuracy across clients, whereas Andante shows larger variability, reflecting their distinct search strategies. ID3 achieves accuracies ranging from 0.40 to 0.76 across client partitions, but only reaches moderate performance in the centralized setting (0.60). Their effectiveness is limited by the propositionalization step, which flattens relational structure into features and prevents trees from exploiting the richer

Table 3
Predictive Accuracy of FILP (Andante and Popper), Federated Decision Tree, and Centralized Popper, Andante and Decision Tree the Post-Intubation Complications Dataset

Client	FILP(Popper)	FILP(Andante)	Decision Tree
Client 1	0.50	0.50	0.56
Client 2	0.75	0.75	0.76
Client 3	0.67	0.56	0.44
Centralized Learning	0.60	0.40	0.60

semantics available to ILP. These results indicate that FILP can match or surpass centralized ILP in accuracy, while federated aggregation does not degrade model quality.

4.4.2. Qualitative Analysis

Decision trees are often considered interpretable because their feature-split structure yields human-readable rules (e.g., `IF arret = 0 AND hypotension = 0 THEN complication = 0`). However, these rules remain shallow: they operate on tabular features, optimize impurity reduction, and lack alignment with domain semantics. In contrast, ILP induces relational rules in first-order logic that capture clinically meaningful interactions, such as:

```
complication(X) :- diabetes(X), shortneck(X), intubation_duration(X,T).
complication(X) :- coronary_artery_disease(X), elderly(X).
```

Crucially, ILP allows for the incorporation of domain-specific background knowledge directly into the learning process, which is infeasible for traditional learners. For instance, the following rules encode clinical facts: if a patient's LEMON score S is greater than or equal to 5, the intubation is expected to be difficult; likewise, if the Cormack score S is greater than 3, a difficult laryngoscopy is expected.

```
hard_intubation(X) :- lemonscore(X,S), S >= 5.
hard_laryngo(X) :- cormack(X,S), S >= 3.
```

Such rules can serve as constraints or inductive biases, enabling the model to generalize from structured medical concepts rather than from feature statistics alone.

While both decision trees and ILP produce readable models, ILP offers deeper plausibility by combining relational reasoning with explicit domain knowledge. FILP extends this capacity to federated settings, preserving interpretability and explainability properties essential for adoption in healthcare. An interesting direction for future evaluation is to involve medical experts to assess whether the induced rules (from ILP) or paths (from ID3) are more interpretable and clinically useful in practice.

4.5. Results on Additional Benchmarks

To complement the medical case study, we extended our evaluation to three established ILP benchmarks: Alzheimer, Carcinogenesis, and Mutagenesis. As preliminary experiments indicated that Popper yields slightly more stable and accurate results than Andante, we report the following benchmark results using Popper as the ILP engine within FILP.

4.5.1. Explainability Metrics

Descriptive Accuracy. Figure 2 reports descriptive accuracy (fidelity) per client, averaged across clients, and compared to the centralized baseline.

Across datasets, client-level fidelity remains consistently high. For example, in Alzheimer (0.67–0.79) and Carcinogenesis (0.63–0.75), local models capture the training distribution with strong alignment

to domain semantics. Intubation shows near-perfect fidelity (0.87–0.97), indicating that rules almost completely explain the observed cases. Mutagenesis, by contrast, remains more difficult, with fidelity around 0.50 across clients, reflecting its higher relational complexity.

Importantly, the mean fidelity across clients is never lower than the centralized baseline (e.g., 0.73 vs. 0.63 in Alzheimer, 0.70 vs. 0.60 in Carcinogenesis), showing that decentralized learning preserves, and in some cases enhances, the ability of ILP to faithfully describe training data.

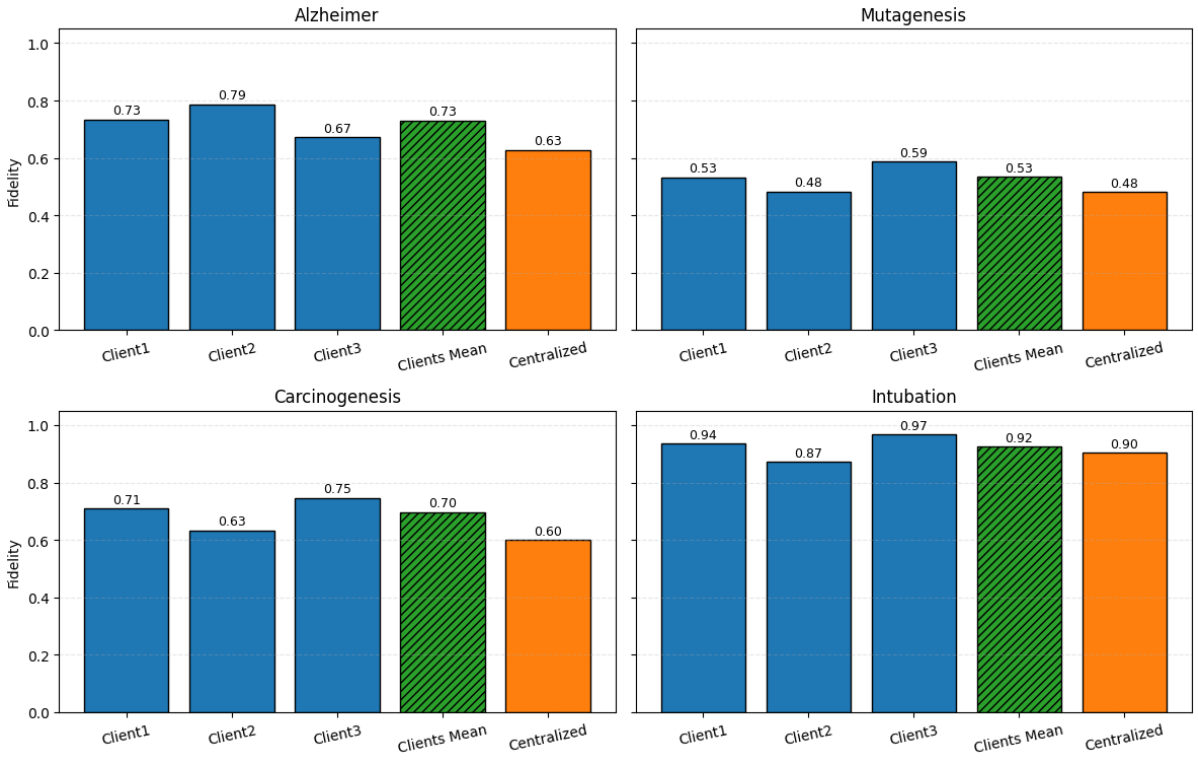


Figure 2: Descriptive Accuracy (Fidelity) across datasets: ILP per Client vs Clients Mean Mean vs Centralized

Rule Complexity. Table 4 compares the model complexity of FILP (federated ILP), centralized Popper, and federated ID3. The comparison is based on two structural metrics: the number of rules and the average rule length.

- For ILP systems (FILP/Popper), the number of rules corresponds to the number of induced clauses, while the average rule length measures the number of body literals per clause. These values are bounded by the declarative bias parameters (`max_clauses`, `max_body`).
- For ID3, the number of rules equals the number of leaves, and the average rule length is the average depth of root–to–leaf paths. This makes the tree metrics functionally comparable to ILP parameters: leaves correspond to `max_clauses`, while path depth parallels `max_body`.

Table 4 highlights three findings. First, FILP induces fewer and shorter rules than centralized Popper (e.g., Carcinogenesis: 12 vs. 27 clauses), showing that federated partitioning regularizes ILP instead of inflating it. Second, compared to federated ID3, FILP achieves greater compactness: while ID3 often yields slightly shorter paths, it does so at the cost of generating more rules (e.g., Mutagenesis: 2.67 vs. 7.33).

Finally, FILP's relational rules retain semantic alignment with domain knowledge, making them more interpretable than the propositional splits of decision trees.

Dataset	FILP		Popper (Centralized ILP)		ID3 (Federated)	
	num_rules	rule_len	num_rules	rule_len	num_rules	rule_len
Alzheimer	18.33	6.92	27.0	7.56	16.67	6.66
Mutagenesis	2.67	3.33	3.0	3.33	7.33	2.97
Carcinogenesis	12.00	6.65	27.0	8.78	9.67	3.86
Intubation	6.67	1.97	15.0	2.80	3.33	1.87

Table 4
Comparison of model complexity across FILP, centralized Popper, and federated ID3. Federated values are averaged across clients; centralized values are computed from a single model trained on the full dataset.

4.5.2. Predictive Accuracy

Figure 3 reports predictive accuracy for FILP across four datasets, comparing per-client performance, the federated mean, and the centralized baseline. FILP achieves accuracies in the federated setting that are comparable to, and in some cases higher than, centralized training. For instance, on Alzheimer (0.60 vs. 0.54) and Intubation (0.64 vs. 0.60), the federated mean exceeds the centralized baseline, showing that distributed ILP does not compromise predictive utility.

Accuracy varies across clients, particularly in Carcinogenesis (0.35–0.62), reflecting heterogeneity in local partitions. Aggregation by majority voting mitigates this variability, producing federated accuracies that remain close to the centralized baseline. Larger datasets (Alzheimer, Carcinogenesis) provide more stable estimates, whereas smaller datasets (Mutagenesis, Intubation) exhibit higher variance due to limited test sizes.

Overall, these results suggest that ILP retains predictive robustness under federated constraints, with federated accuracies closely matching or slightly exceeding centralized baselines. However, variability across clients, particularly in smaller datasets, indicates that dataset scale and partitioning strongly affect generalization performance.

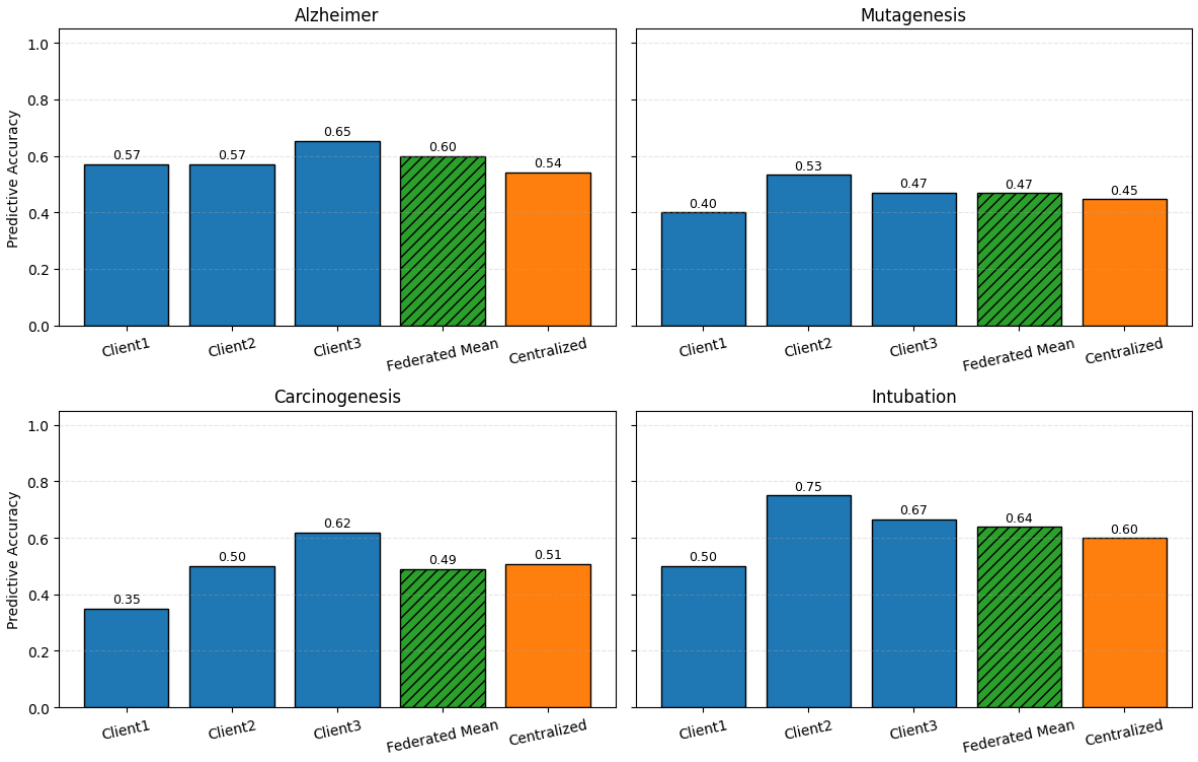


Figure 3: Predictive accuracy across datasets: centralized learning vs. mean client accuracy in the federated setting of FILP.

5. Conclusion

In this work, we introduced FILP, a novel framework that integrates Inductive Logic Programming into Federated Learning to enable symbolic, interpretable, and privacy-preserving model induction. FILP allows each client to learn logic-based rules from local data using ILP systems such as Andante (Progol-based) and Popper (ASP-based), and combines these symbolic models via majority voting to derive a global consensus theory.

Our evaluation on a real-world clinical task illustrates the trade-offs between symbolic and statistical learners. Decision trees achieve competitive predictive performance, but their effectiveness is constrained by the propositionalization step and their lack of semantic alignment with domain knowledge. We chose decision trees as a baseline because they are the most established interpretable, rule-based machine learning model, providing a natural comparison point for ILP in terms of both predictive power and explanation style. ILP methods, in contrast, yield relational rules that align with clinical semantics and domain knowledge, offering transparency and explanatory depth. Such models are naturally aligned with expert validation and reasoning, a critical requirement in high-stakes fields like medicine.

FILP represents a step toward trustworthy and interpretable federated AI. The central contribution of this study is to investigate whether ILP can be effectively integrated within a federated setting, and whether this integration provides tangible benefits in terms of accuracy and interpretability. Interestingly, our results indicate that in some cases federated ILP performs better than centralized ILP, which may be explained by symbolic learners generalizing more effectively from smaller, structurally coherent subsets. In contrast, centralized aggregation may introduce heterogeneity that impairs the induction of compact logical rules.

As future work, we aim to (i) broaden the evaluation to further ILP benchmarks beyond the ones already considered, including comparisons against other rule-based and explainable learners beyond decision trees, (ii) investigate symbolic aggregation strategies beyond majority voting, (iii) explore richer integration of clinician-defined background knowledge to further enhance semantic fidelity and generalization, and to evaluate potential gains in predictive accuracy, and (iv) assess the scalability and robustness of FILP on larger and more diverse datasets.

Acknowledgments

The authors thank the University of Namur for its support and the Walloon Region for partial support through the Ariac project (convention 210235) and the CyberExcellence project (convention 2110186). Special thanks go to Dr. Khaled Ouertani for sharing the medical dataset.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to perform grammar and spelling checks. After using this tool, the author(s) reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] E. J. Topol, High-performance medicine: the convergence of human and artificial intelligence, Nature medicine 25 (2019) 44–56.

[2] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, et al., Scalable and accurate deep learning with electronic health records, NPJ digital medicine 1 (2018) 18.

[3] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, nature 542 (2017) 115–118.

[4] N. Gahlan, D. Sethia, Federated learning in emotion recognition systems based on physiological signals for privacy preservation: a review, Multimedia Tools and Applications (2024) 1–69.
[5] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
[6] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. Galtier, B. Landman, K. H. Maier-Hein, et al., The future of digital health with federated learning, NPJ Digital Medicine 3 (2020) 1–7.
[7] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Processing Magazine 37 (2020) 50–60.
[8] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, et al., Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data, in: Scientific Reports, volume 10, Nature Publishing Group, 2020, pp. 1–12.
[9] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, Journal of Healthcare Informatics Research 5 (2021) 1–19.
[10] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.
[11] S. R. Heiyanthuduwage, I. Altas, M. Bewong, M. Z. Islam, O. B. Deho, Decision trees in federated learning: Current state and future opportunities, IEEE Access (2024).
[12] A. Holzinger, C. Biemann, C. S. Pattichis, D. B. Kell, What do we need to build explainable ai systems for the medical domain?, arXiv preprint arXiv:1712.09923 (2017).
[13] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a "right to explanation", AI Magazine 38 (2017) 50–57.
[14] S. Muggleton, Inductive logic programming, New generation computing 8 (1991) 295–318.
[15] A. Cropper, S. Dumančić, S. H. Muggleton, Turning 30: New ideas in inductive logic programming, arXiv preprint arXiv:2002.11002 (2020).
[16] R. D. King, A. Srinivasan, Prediction of rodent carcinogenicity bioassays from molecular structure using inductive logic programming., Environmental Health Perspectives 104 (1996) 1031–1040.
[17] R. D. King, S. Muggleton, R. A. Lewis, M. Sternberg, Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase., Proceedings of the national academy of sciences 89 (1992) 11322–11326.
[18] A. Srinivasan, R. D. King, S. H. Muggleton, M. J. Sternberg, Carcinogenesis predictions using ilp (1997) 273–287.
[19] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, IEEE Transactions on Knowledge and Data Engineering 35 (2019) 3347–3366.
[20] B. Yuan, S. Ge, W. Xing, A federated learning framework for healthcare iot devices, arXiv preprint arXiv:2005.05083 (2020).
[21] M. H. U. Rehman, W. Hugo Lopez Pinaya, P. Nachev, J. T. Teo, S. Ourselin, M. J. Cardoso, Federated learning for medical imaging radiology, The British Journal of Radiology 96 (2023) 20220890.
[22] A. Qayyum, K. Ahmad, M. A. Ahsan, A. Al-Fuqaha, J. Qadir, Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge, IEEE Open Journal of the Computer Society 3 (2022) 172–184.
[23] J. Li, Y. Meng, L. Ma, S. Du, H. Zhu, Q. Pei, X. Shen, A federated learning based privacy-preserving smart healthcare system, IEEE Transactions on Industrial Informatics 18 (2021).
[24] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, G. Groh, Shap-based explanation methods: a review for nlp interpretability, in: Proceedings of the 29th international conference on computational linguistics, 2022, pp. 4593–4603.
[25] N. K. Alshammari, A. A. Alhusaini, A. Pasha, S. S. Ahamed, T. R. Gadekallu, M. Abdullah-Al-Wadud,

R. A. Ramadan, M. H. Alrashidi, Explainable federated learning for enhanced privacy in autism prediction using deep learning, Journal of Disability Research 3 (2024) 20240081.
[26] D. Janzing, L. Minorics, P. Blöbaum, Feature relevance quantification in explainable ai: A causal problem, in: International Conference on artificial intelligence and statistics, PMLR, 2020, pp. 2907–2916.
[27] C. Xu, G. Chen, C. Li, Federated learning for interpretable short-term residential load forecasting in edge computing network, Neural Computing and Applications 35 (2023) 8561–8574.
[28] S. N. Zeleke, M. Bochicchio, Towards explainable federated learning in healthcare: A study on heart arrhythmia detection (2024).
[29] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature machine intelligence 1 (2019) 206–215.
[30] P. Xing, S. Lu, H. Yu, Federated neuro-symbolic learning, arXiv preprint arXiv:2308.15324 (2023).
[31] M. Billa, Symbolic regression for medical scoring systems: a bayesian and multi-objective approach, in: Sistemi Evoluti per Basi di Dati, 2024. URL: https://api.semanticscholar.org/CorpusID:271866221.
[32] J. A. Scott, H. Zakerinia, C. Lampert, Pefll: Personalized federated learning by learning to learn, in: 12th International Conference on Learning Representations, 2024.
[33] Q. Li, W. Zhaomin, Y. Cai, C. M. Yung, T. Fu, B. He, et al., Fedtree: A federated learning system for trees, Proceedings of Machine Learning and Systems 5 (2023).
[34] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM workshop on artificial intelligence and security, 2019, pp. 1–11.
[35] T. Dong, S. Li, H. Qiu, J. Lu, An interpretable federated learning-based network intrusion detection framework, arXiv preprint arXiv:2201.03134 (2022).
[36] K. Gupta, P. Kumar, S. Upadhyaya, M. Poriye, S. Aggarwal, Fuzzy logic and machine learning integration: Enhancing healthcare decision-making, International Journal of Computer Information Systems and Industrial Management Applications 16 (2024) 20–20.
[37] G. Castellano, R. Scaringi, G. Vessio, G. Zaza, et al., Integrating graph neural networks and fuzzy logic to enhance deep learning interpretability, in: Proceedings of the Fourth International Workshop on Multilingual Semantic Web (MSW 2024), Paris, France, 2024, pp. 11–13.
[38] D. Połap, Fuzzy consensus with federated learning method in medical systems, IEEE Access 9 (2021) 150383–150392.
[39] V. Srivastava, V. Lamba, V. S. Mathada, C. Bulla, N. Gupta, P. Veeramanikandan, et al., An iot-based framework employing fuzzy logic and federated learning for decentralized decision-making, International Journal of Information Technology (2025) 1–7.
[40] W. Jiang, Y. Zhang, H. Han, X. Liu, J. Gwak, W. Gu, A. Shankar, C. Maple, Fuzzy ensemble-based federated learning for eeg-based emotion recognition in internet of medical things, Journal of Industrial Information Integration 44 (2025) 100789.
[41] A. Wilbik, P. Grefen, Towards a federated fuzzy learning system, 2021.
[42] P. Ducange, F. Marcelloni, A. Renda, F. Ruffini, Federated learning of xai models in healthcare: a case study on parkinson's disease, Cognitive Computation 16 (2024) 3051–3076.
[43] J. L. C. Bárcena, P. Ducange, F. Marcelloni, A. Renda, Increasing trust in ai through privacy preservation and model explainability: Federated learning of fuzzy regression trees, Information Fusion 113 (2025) 102598.
[44] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery., Queue 16 (2018) 31–57.
[45] S. Muggleton, L. De Raedt, Inductive logic programming: Theory and methods, The Journal of Logic Programming 19 (1994) 629–679.
[46] A. Cropper, R. Morel, Learning programs by learning from failures, Machine Learning 110 (2021) 801–856.
[47] A. Cropper, Forgetting to learn logic programs, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 3676–3683.

[48] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, S. J. Gershman, Building machines that learn and think like people, Behavioral and brain sciences 40 (2017).
[49] I. Bratko, S. Muggleton, Applications of inductive logic programming, Communications of the ACM 38 (1995) 65–70.
[50] P. Sen, B. W. de Carvalho, R. Riegel, A. Gray, Neuro-symbolic inductive logic programming with logical neural networks, in: Proceedings of the AAAI conference on artificial intelligence, volume 36, 2022, pp. 8212–8219.
[51] Z. Zhang, L. Yilmaz, B. Liu, A critical review of inductive logic programming techniques for explainable ai, IEEE transactions on neural networks and learning systems 35 (2023) 10220–10236.
[52] T. Dreossi, et al., Bridging deep learning and logic programming for explainability through ilp, ELECTRONIC PROCEEDINGS IN THEORETICAL COMPUTER SCIENCE 416 (2025) 314–323.
[53] Z. Ma, Y. Zhuang, P. Weng, H. H. Zhuo, D. Li, W. Liu, J. Hao, Learning symbolic rules for interpretable deep reinforcement learning, arXiv preprint arXiv:2103.08228 (2021).
[54] B. Wolfson, E. Acar, Differentiable inductive logic programming for fraud detection, arXiv preprint arXiv:2410.21928 (2024).
[55] A. Siromoney, L. Raghuram, A. Siromoney, I. Korah, G. Prasad, Inductive logic programming for knowledge discovery from mri data, IEEE Engineering in Medicine and Biology Magazine 19 (2000) 72–77.
[56] S. Muggleton, Inverse entailment and progol, New Generation Computing 13 (1995) 245–286.
[57] J. Cussens, Part-of-speech tagging using progol, in: International conference on inductive logic programming, Springer, 1997, pp. 93–108.
[58] C. Helma, T. Cramer, S. Kramer, L. De Raedt, Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds, Journal of chemical information and computer sciences 44 (2004) 1402–1411.
[59] A. Cropper, S. H. Muggleton, Learning efficient logical robot strategies involving composable objects, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 2015, pp. 3423–3429.
[60] A. Srinivasan, Mutagenesis: Ilp experiments in a non-determinate biological domain, in: Proc. of the Fourth International Workshop on Inductive Logic Programming (ILP'94), 1994.
[61] R. D. King, M. J. Sternberg, A. Srinivasan, Relating chemical activity to structure: an examination of ilp successes, New Generation Computing 13 (1995) 411–433.
[62] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, H. L. Kwing, T. Parcollet, P. P. d. Gusmão, N. D. Lane, Flower: A friendly federated learning research framework, arXiv preprint arXiv:2007.14390 (2020).

A. Dataset Partitions

Dataset partitions used for federated training and evaluation. Each dataset is first split into:

- **Global split:** 80% training vs. 20% global test (e.g., `complication80`/`complication20`, `muta80`/`muta20`).
- **Client folds:** the 80% training portion is further divided into three client folds (e.g., `complication1--3`, `mutagenesis1--3`, `carcino1--3`, `alzheimer1--3`).
- **Local tests:** The global test set is partitioned into three disjoint subsets, yielding one local test set per client (e.g., `testcom1--3`, `mutagentest1--3`, `test1--3`, `testalzheimer1--3`).

This organization ensures stratified splits for both federated clients and global evaluation.

(a) Post-Intubation Complications Dataset Partitions

Partition	Positives	Negatives	Total
complication	72	46	118
complication20	15	10	25
complication80	57	36	93
complication1	19	12	31
complication2	19	12	31
complication3	19	12	31
testcom1	5	3	8
testcom2	5	3	8
testcom3	5	4	9

(b) Alzheimer Acetyl Dataset Partitions

Partition	Positives	Negatives	Total
alzheimer_acetyl	530	530	1060
alzheimer20	106	106	212
alzheimer80	424	424	848
alzheimer1	141	141	282
alzheimer2	141	141	282
alzheimer3	142	142	284
testalzheimer1	35	35	70
testalzheimer2	35	35	70
testalzheimer3	36	36	72

(c) Carcinogenesis Dataset Partitions

Partition	Positives	Negatives	Total
carcinogenesis	162	136	298
car20	33	28	61
car80	129	108	237
carcino1	43	36	79
carcino2	43	36	79
carcino3	43	36	79
test1	11	9	20
test2	11	9	20
test3	11	10	21

(d) Mutagenesis Dataset Partitions

Partition	Positives	Negatives	Total
mutagenesis	138	92	230
muta20	28	19	47
muta80	110	73	183
mutagenesis1	36	24	60
mutagenesis2	36	24	60
mutagenesis3	38	25	63
mutagentest1	9	6	15
mutagentest2	9	6	15
mutagentest3	10	7	17

Figure 4: Data distribution across partitions for the four datasets (Post-Intubation Complications, Alzheimer, Carcinogenesis, Mutagenesis).