Formal Logic Enabled Personalized Federated Learning through Property Inference

Ziyan An, Taylor T. Johnson, Meiyi Ma

Department of Computer Science, Vanderbilt University, Nashville, TN, USA
{ziyan.an, taylor.johnson, meiyi.ma}@vanderbilt.edu

Abstract

Recent advancements in federated learning (FL) have greatly facilitated the development of decentralized collaborative applications, particularly in the domain of Artificial Intelligence of Things (AIoT). However, a critical aspect missing from the current research landscape is the ability to enable data-driven client models with symbolic reasoning capabilities. Specifically, the inherent heterogeneity of participating client devices poses a significant challenge, as each client exhibits unique logical reasoning properties. Failing to consider these device-specific specifications can result in missing critical properties in the client predictions, leading to suboptimal performance. This work proposes a new training paradigm that leverages temporal logic reasoning to address this issue. Our approach enhances the training process by incorporating mechanically generated logic expressions for each FL client. Additionally, we develop aggregation clusters and a partitioning algorithm to effectively group clients based on the alignment of their temporal reasoning properties. We evaluate the proposed method on two tasks: a real-world traffic volume prediction task consisting of sensory data from fifteen states and a smart city multi-task prediction utilizing synthetic data. The evaluation results exhibit clear improvements, with performance accuracy improved by up to 54% across all sequential prediction models.

Introduction

In recent years, modern Artificial Intelligence (AI) models have been adapted to handle massively-distributed tasks deployed on non-independent and identically distributed (i.i.d) devices, including AI-empowered Internet-of-Things services such as smart cities and smart healthcare (Nguyen et al. 2021; Ma et al. 2019; Ma, Stankovic, and Feng 2021; Preum et al. 2021). However, a critical challenge that remains unsolved is the ability to enable large-scale distributed data-driven models with *symbolic reasoning* capabilities; that is, incorporating logical properties to guide and enhance the learning process.

More specifically, federated learning (FL) (McMahan et al. 2017) frameworks have been developed to achieve remarkable performance for distributed training with reduced privacy risks. These frameworks enable collaborations between participating devices by aggregating their models through a centralized moderator. FL models have proven particularly useful in privacy-sensitive outdoor sensor networks, where participating clients can receive non-i.i.d or heterogeneous input data. Client data is securely maintained on-device, ensuring that sensitive information remains local and private. To facilitate model updates and collaboration, only client model parameters are synchronized with the centralized moderator.

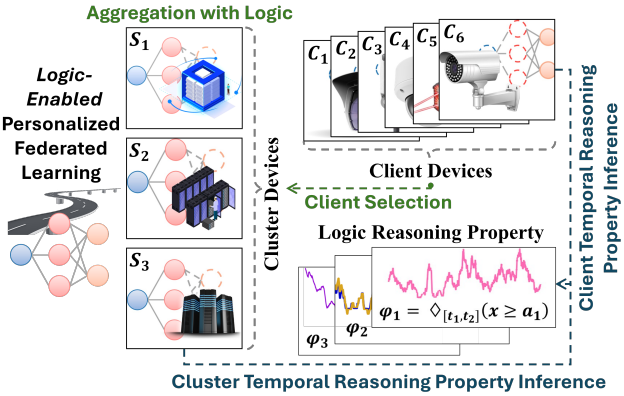

Figure 1: FedSTL consists of $\mathcal{S}$ cluster devices, and $\mathcal{C}$ client devices. During each communication round, client models are partitioned into clusters. Then, inferred temporal reasoning properties are enhanced on predictive models.

However, dealing with client heterogeneities symbolically remains a challenge for FL models. Previous FL designs have attempted to address this issue through methods such as client selection, clustering, and regularization (Collins et al. 2021; Li et al. 2021a,b; Mohri, Sivek, and Suresh 2019; Yu, Bagdasaryan, and Shmatikov 2020). Unfortunately, these methods fail to provide any reasoning for the model's outputs. Furthermore, no logic reasoning properties or domain-specific knowledge can be integrated and enforced through such approaches.

Consider a sequential real-world prediction task that involves multiple types of sensors (e.g., radar sensors, video detection systems, air quality monitors) deployed at various

Code implementation is available at https://github.com/AICPS-Lab/FedSTL.git.

locations throughout a city (e.g., highways, town roads, indoors) (FHWA 2016; Ma et al. 2021a; Ma, Stankovic, and Feng 2018; Ma, Preum, and Stankovic 2017). The heterogeneity of deployment locations and sensor types leads to monitoring data with diverse distributions, with each client following their unique time-series patterns. Previous FL approaches have failed to provide any symbolic reasoning capabilities that enable client predictions to follow these distinct time-series patterns.

Additionally, current FL frameworks lack the ability of *intra-task symbolic reasoning*. In the case of the previous example, multi-class prediction tasks, such as road occupancy and noise level, can exhibit correlations, but this correlation may not be consistent across all clients. A more advanced FL framework should incorporate the ability to understand how diverging intra-task logical reasoning patterns as such can be aligned with client predictions.

Motivated by these challenges, we present a new FL framework towards distributed temporal symbolic reasoning dubbed *FedSTL*, short for *signal temporal logic-enabled personalized federated learning*. Figure 1 depicts the overall structure of our approach. In FedSTL, the task is to predict future multivariate traces based on local datasets for each client. We aim to tackle two tasks in FL: 1) identifying and regulating client models with localized temporal reasoning properties at the training time, and 2) aggregating each client with others with similar properties.

We use the term "property" to refer to any prior domain knowledge, world knowledge, or expertise knowledge. However, manually specifying comprehensive multivariate properties for large-scale FL systems is another significant challenge, where each client differs from the others. Hence, our model enables extracting localized client knowledge *automatically* at training time to address this issue. By properties and specifications, we refer to patterns derived from the private data of each client. In Table 1, we provide a more comprehensive set of examples demonstrating the types of properties and specifications that can seamlessly integrate into our framework.

In the personalized FL framework, each client has a locally distinct model responsible for making predictions based on their private input data (Chen et al. 2022). The central aggregator(s) then leverage updates from multiple client models to improve the central model. However, in our framework, we introduce two modifications to this popular learning paradigm. Firstly, we incorporate client prediction regulations with automatically inferred logic reasoning. Secondly, we use temporal logic properties to cluster clients based on the alignment of these properties. Within each cluster, the members are aggregated to contribute to the same shared model. Additionally, our framework ensures *multi-granularity personalization* by enforcing specialized cluster and client properties.

By regulating client predictions with locally inferred properties, we enhance these properties for each client using a teacher-student learning paradigm, aligning the prediction results more closely with the specified requirements. This enables us to effectively address the challenge posed by heterogeneity in client deployment locations and sensor types, which leads to different data distributions for each client. In addition, we employ clustering based on property alignments, allowing us to aggregate clients with similar properties. With this approach, we group clients whose properties exhibit higher similarity, thus reducing the aggregation of weights from heterogeneous clients. Throughout this work, we use the terms "property" and "specification" interchangeably. We summarize the contributions as follows:

- FedSTL is a novel personalized FL framework that enhances temporal reasoning through automatically inferred logic properties for heterogeneous FL clients.
- Our framework is designed to facilitate the automatic discovery and induction of client and cluster temporal logic specifications from datasets.
- FedSTL clusters FL clients based on agreements of specifications, enabling cross-client collaboration for cluster models to exploit shared knowledge.
- We evaluate FedSTL under various dataset settings, including two realistic testing scenarios with both real-world data and simulated datasets. Empirical evaluations demonstrate that FedSTL improves client-level model property satisfaction while boosting prediction accuracy compared to state-of-the-art baseline frameworks.

Notations $\{\mathcal{C}_i\}$: client models; $\{\mathcal{S}_j\}$: cluster models; $\{\mathcal{X} \times \mathcal{Y}\} \in \mathcal{D}_i$: client datasets; θ: model parameters; F: local objective function; G: aggregation function; $\Box$ *always*; $\Diamond$: *eventually*; $\mathcal{U}$: *until*; μ: predicate variable; φ: STL formula; $\varphi(\alpha)$: templated STL formula; ρ: STL robustness.

Problem Formulation

A general FL framework consists of a central server and $\{\mathcal{C}_i\}$ client devices, each with its own local dataset $\mathcal{D}_i$ consisting of i.i.d data points (x, y). The primary objective is to train a central model parameterized by θ_g without requiring the clients to share their private data.

During each communication round, the central server first broadcasts the current version of θ_g to the participating clients. The clients then use local datasets $\mathcal{D}_i$ to train the model for a certain number of iterations. Then, the clients send the updated model parameters $\{\theta_i\}, i \in \mathcal{C}$ back to the central server, which performs an aggregation of these updates to generate a new version of θ_g. This process is repeated for multiple times.

In this work, our focus is on training personalized parametric models $\{\theta_i\}$ for clients using a slightly different setup. More concretely, we partition client models into $\{\mathcal{S}_j\}$ clusters, each with its own model parameter θ_j. During each communication round, the cluster models $\{\theta_j\}$ are broadcast to the clients assigned to the corresponding cluster. Clusters and clients then follow a similar back-and-forth communication rule as other general FL training paradigms.

Let $f(\theta; x, y) \rightarrow \mathbb{R}$ denote the loss of model θ at the data point (x, y). For each client $i \in \mathcal{C}$, let $F_i(\theta_i) \coloneqq \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[f(\theta_i; x, y)]$ be the local objective function. Our goal is to obtain better client models $\{\hat{\theta}_i\}$ that are close to the optimal models $\theta_i^* \in \text{argmin}_{\theta_i} F_i(\theta_i)$ for each $i \in \mathcal{C}$. Additionally, we use $G(\cdot)$ to denote an aggregation function

Temporal Reasoning Property	Templated Logic Formula	Parameters
1: Operational Range: Signal is upper-bounded by threshold a and lower-bounded by threshold b.	$\bigwedge_{i=1}^{1,2,\ldots,\tau}(\square_{[i,i+t]}(x \leq a_i \wedge x \geq b_i))$	a_i, b_i
2: Existence: Signal should eventually reach the upper extreme a and the lower extreme b.	$\bigwedge_{i=1}^{1,2,\ldots,\tau}(\Diamond_{[i,i+t]}(x \leq a_i \wedge x \geq b_i))$	a_i, b_i
3: Until: Signal must satisfy one specification at all times until another condition is met.	$\bigwedge_{i=1}^{1,2,\ldots,t}((x < a_i)\,\mathcal{U}_{[i,i+1]}(x < b_i))$	a_i, b_i
4: Intra-task Reasoning: The difference between signal variables x_1 and x_2 should be greater than a.	$\bigwedge_{i=1}^{1,2,\ldots,\tau}(\square_{[i,i+t]}((x_1 - x_2) > a_i))$	a_i
5: Temporal Implications: The happening of one event indicates that another event will happen at some point in the future.	$\square_{[t_1,t_2]}((x \geq a_1) \rightarrow \Diamond_{[t_3,t_4]}(x \geq a_2))$	a_1, a_2
6: Intra-task Nested Reasoning: The signal variable x_1, when greater than a threshold a, indicates x_2 will eventually reach a threshold b.	$\square_{[t_1,t_2]}((x_1 \geq a) \rightarrow \Diamond_{[t_3,t_4]}(x_2 \geq b))$	a, b
7: Multiple Eventualities: Multiple events must eventually happen, but their order can be arbitrary.	$\Diamond_{[t_1,t_2]}(x \geq a_1) \wedge \cdots \wedge \Diamond_{[t_3,t_4]}(x \geq a_n)$	$a_1, a_2 \cdots a_n$
8: Template-free: specification mining without a templated formula.	No pre-defined templates are needed.	n/a.

Table 1: Examples of temporal reasoning templates specified with STL.

that defines how the client model updates are combined to form the global model update.

Temporal Reasoning Property Inference

Temporal Logic Specification

We first introduce the preliminaries of signal temporal logic (STL) (Maler and Nickovic 2004), which is a formalism that provides a flexible and rigorous way to specify temporal logic reasoning. To begin, we provide the syntax of an STL formula, as defined in Definition 1.

Definition 1 (STL syntax)**.**

$$\varphi ::= \mu \mid \neg\mu \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2$$
$$\mid \Diamond_{[a,b]}\varphi \mid \square_{[a,b]}\varphi \mid \varphi_1 \mathcal{U}_{[a,b]}\varphi_2$$

We use the notation $[a, b] \in \mathbb{R}_{\geq 0}$, with $a \leq b$, to represent a temporal range. Let $\mu : \mathbb{R}^n \rightarrow \{\top, \bot\}$ be a signal predicate (e.g. $f(x) \geq 0$) on the signal variable $x \in \mathcal{X}$. Additionally, we refer to different STL formulas using φ, φ_1, and φ_2. We use $\square$ to denote the property "always," which requires the formula φ to be true at all *future* time steps within $[a, b]$. Similarly, we use $\Diamond$ to denote "eventually," which requires the formula φ to be true at some future time steps between $[a, b]$. Finally, we use $\mathcal{U}$ to denote "until," which specifies that φ_1 is true until φ_2 becomes true.

An example of an STL formula is $\square_{[0,5]}((x_1 \geq 0.75) \rightarrow (x_2 \geq 10))$, which formally specifies that if the signal variable x_1 exceeds or equals to 0.75 during future times $[0, 5]$, then the signal variable x_2 should always be greater than or equal to 10.

Logic Inference Through Observed Data

Logic inference Logic inference (Bartocci et al. 2022) is the process of generating logic properties based on observed facts when the desired system property is unknown or only partially available. Given some prior knowledge about the possible form of a logic property, the logic inference algorithm (specification mining (Jha et al. 2017)) learns the complete logic formula. Formally, the logic property inference task is described in Definition 2.

Definition 2 (STL property inference)**.** Given an observed fact x and a templated STL formula $\varphi(\alpha)$, where α is an unknown parameter, the task is to find a value for α such that φ is satisfied for all instances of x.

We provide a practical example of STL property inference in Example 1.

Example 1 (An example of STL inference)**.** Given a templated STL property $\varphi_k(\alpha) = \square_{[0,5)}((x_1 \geq 0.75) \rightarrow (x_2 \geq \alpha))$, the goal is to find a value for the unknown parameter α such that during future time stamps 0 to 5, if the signal variable x_1 exceeds or equals to 0.75, the signal variable x_2 should always be greater than or equal to α.

Furthermore, we have summarized eight categories of temporal reasoning properties in Table 1 that can be expressed using STL and can be inferred through specification mining algorithms.

In practice, there are infinite possible values that a free parameter (e.g., α) can take to make a templated formula $\varphi(\alpha)$ valid. However, not all valid values of α are equal in terms of enhancing the reasoning property during the training process. To be more specific, we need to find an α value such that the observed facts satisfy $\varphi(\alpha)$ with a small margin. In the property inference process, the STL quantitative semantics (robustness) as defined in (Donzé and Maler 2010) is used as a real-valued measurement for property satisfaction.

In the following example, we briefly show how to utilize the robustness value as a real-valued measurement for property satisfaction. In this instance, we provide a 5-step sequential data, which is then evaluated

against the temporal property listed in Example 1. Continuing with the templated STL defined in Example 1, when $\alpha = 10$, the 5-step sequential data $\mathcal{X} = ((0.25, 20), (0.25, 18), (0.5, 16), (0.6, 14), (0.75, 12))$ results in a robustness value of $\rho(\varphi, \mathcal{X}) = -2$. The resulting robustness value is -2, indicating that the property was not satisfied. However, a value close to $\alpha = 12$ would fit $\mathcal{X}$ tightly, resulting in an STL property that more accurately describes the true observation. Therefore, the STL inference task can be better formulated as Definition 3, which incorporates the notion of a tight bound.

Definition 3 (STL property inference with a tight bound)**.** Given an observed data $\mathcal{X}$ and a templated STL formula $\varphi(\alpha)$, where α is an unknown parameter. The goal is to identify a value for α that results in a tightly-fitted logic property, expressed by the equation $\rho(\varphi, \mathcal{X}; \alpha) = \epsilon$, where a smaller positive ϵ indicates a closer alignment with the data.

The task defined in Definition 3 can be effectively solved using the following algorithm, which can be solved with either gradient-free or gradient-based numerical optimization methods (Jha et al. 2017).

$$\begin{aligned} &\min |\epsilon| \ s.t. \ \epsilon = p' - p \\ &\text{where } \rho(\varphi(p), \mathcal{X}, t) \geq 0 \text{ and } \rho(\varphi(p'), \mathcal{X}, t) < 0 \end{aligned} \tag{1}$$

In the templated logic formula φ, let p and p' denote candidate values of free parameters, and let t be a timestamp. our objective is to minimize the value of $|\epsilon|$, which represents the discrepancy between a satisfactory parameter value and an unsatisfactory parameter value. The STL robustness function is not differentiable at zero. Therefore, to tackle this issue, alternatives such as the "tightness metric" (Jha et al. 2017) can be employed to effectively address this problem.

Logic-Enabled Federated Learning

Enhancing STL-based Logic Reasoning Property

We consider the workflow illustrated in Figure 2. During each training iteration, FedSTL utilizes Equation 1 to infer a logic reasoning property φ from the client dataset D_i. The inferred property is then incorporated into the client prediction process through an additional loss term $\mathcal{L}_p$, which penalizes the neural network for any deviations from the property. This is achieved by implementing a teacher-student structure (Hinton, Vinyals, and Dean 2015), which will be discussed in detail in subsequent sections.

We first describe how FedSTL framework regulates and corrects client prediction such that it satisfies the logic property $\varphi := \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ extracted by logic inference.

To begin with, recall that any locally inferred logic reasoning property must be satisfied by every data point in the client dataset. Additionally, any signal temporal logic formula can be represented by its equivalent Disjunctive Normal Form (DNF), which typically takes the form of $P \vee Q \vee R \vee \ldots$. Each clause in a DNF formula consists of either variables, literals, or conjunctions, for example, $P := p_1 \wedge \neg p_2 \wedge \ldots \wedge p_n$. Essentially, the DNF form specifies a range of satisfaction where *any* logic clause connected with the disjunction operator satisfies the STL formula φ.

Algorithm 1: CLUSTER_ID: Cluster identity mapping

Parameters: cluster devices $\{\mathcal{S}_j\}$, participating client models $\{\mathcal{C}_i\}$.

1: Initialize client identity mapping $\mathcal{I}$.
`/* generate client data property */`
2: **for** client dataset $\mathcal{D}_i^s$ **do**
3: Generate client data property φ_i^s on dataset $\mathcal{D}_i^s$
4: **end for**
`/* clusters select clients */`
5: **for** cluster device $\mathcal{S}_j$ **do**
6: **for** client data $\mathcal{D}_i^s$ **do**
7: Generate $\hat{\mathcal{Y}}$ with $\mathcal{S}_j$ and $\mathcal{D}_i^s$
8: Calculate empirical logic reasoning loss $\mathcal{L}_p(\varphi_i^s, \hat{\mathcal{Y}})$
9: **end for**
10: Cluster selects clients $\mathcal{C}_i$ with the lowest $\mathcal{L}_p$
11: Append $\mathcal{C}_i$ to identity mapping $\mathcal{I}$.
12: **end for**
13: **return** $\mathcal{I}$ (client identity mapping)

Taking advantage of this fact, our framework leverages the DNF equivalents of automatically generated client properties. We aim to find a logic clause φ^* that represents the closest approximation to the model prediction $\hat{\mathcal{Y}}$ in terms of satisfying the property (Ma et al. 2020). To achieve this, we introduce an additional loss term $\mathcal{L}_p(\varphi^*, \hat{\mathcal{Y}})$ that quantifies the distance between the model prediction and the closest satisfying trace. In practice, we use the L-1 distance as a metric to quantify this loss, capturing the absolute difference between the predicted and desired outputs.

Dynamic Temporal Logic-Based Clustering

One advantage of the FedSTL framework is its dynamic assignment of client models to clusters, which allows the aggregation process to adapt to changes in logic properties as they occur. At a high level, clients with similar temporal reasoning properties are grouped together in clusters, while clients with different properties are not aggregated together.

We demonstrate this process in Algorithm1, which is executed every m communication rounds, where m is a hyperparameter. Let $\mathcal{D}_i^s, i \in \mathcal{C}$ be a small sample of desensitized client data. During each clustering process, FedSTL generates the logic property φ_i^s for each participating client on their respective datasets $\mathcal{D}_i^s$ (line 3, Alg.1). Then, each cluster device $\mathcal{S}_j$ generates predictions $\hat{\mathcal{Y}}$ on the samples in $\mathcal{D}_i^s$ (line 7, Alg.1). Next, we calculate the empirical logic reasoning loss $\mathcal{L}_p(\varphi_i^s, \hat{\mathcal{Y}})$ for each cluster model $\mathcal{S}_j$ on client dataset $\mathcal{D}_i^s$ (line 8, Alg.1). At the end of each round, we select cluster members based on the lowest logic reasoning loss $\mathcal{L}_p$ (line 10, Alg. 1).

Hierarchical Logic Reasoning Strengthening

The pseudocode for the collaborative updating paradigm of FedSTL is presented in Algorithm 2. The framework operates for a total of $\mathcal{T}$ communication rounds, and at each round, participating client models $\{\mathcal{C}_i^{(t)}\}$ are selected based on a pre-defined participation rate r (line 2, Alg. 2). Our

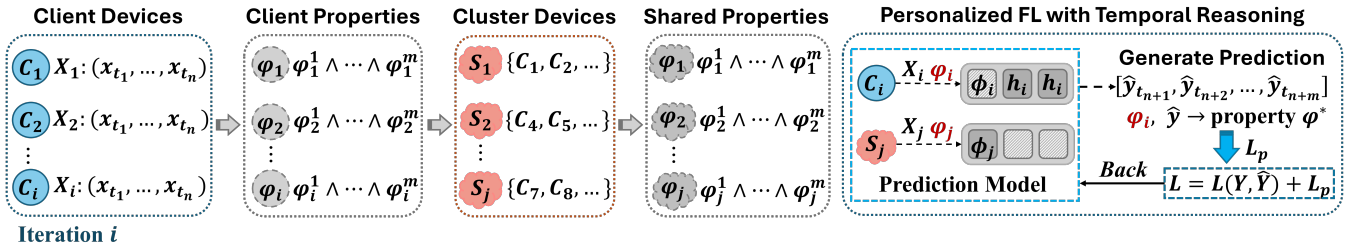

Figure 2: The training workflow for one iteration in the framework involves the inference of client logic properties $\{\varphi_i\}$ and cluster logic properties $\{\varphi_j\}$. Based on the alignment of these properties, clients are partitioned into clusters. Then, our framework enhances personalized FL by incorporating both client and cluster reasoning properties during training.

Algorithm 2: FedSTL: Client federation and update

1: **for** $t = 1, 2, \ldots, \mathcal{T}$ **do**
2: $\{\mathcal{C}_i^{(t)}\} \leftarrow$ Selected clients with participation rate r
3: $\mathcal{I}^{(t)} \leftarrow$ CLUSTER_ID($\{\mathcal{S}_j\}, \{\mathcal{C}_i^{(t)}\}$)
4: Clusters broadcast current model $\phi_j^{(t)}$ to clients
`/* update client models */`
5: **for** client i in $\{\mathcal{C}_i^{(t)}\}$ *in parallel* **do**
6: Client i initializes layers $\theta_i^{(t)}$
7: **for** $t = 1, 2, \ldots, \tau$ **do**
8: $(\theta_i^{(t)}) \leftarrow \text{SGD}(\theta_i^{(t)}, F_i, \eta)$
9: **end for**
10: Client i sends shared $\phi_i^{(t)}$ to cluster $\mathcal{S}_j$
11: **end for**
`/* update cluster models */`
12: **for** cluster j in $\{\mathcal{S}_j^{(t)}\}$ *in parallel* **do**
13: Cluster j performs member aggregation:
$\phi_j^{(t)} \leftarrow \text{G}(\phi_j^{(t)}, \{\phi_i^{(t)}\})$
14: **for** $t = 1, 2, \ldots, \kappa$ **do**
15: $(\phi_j^{(t+1)}) \leftarrow \text{SGD}(\phi_j^{(t)}, F_j, \eta_j)$
16: **end for**
17: **end for**
18: **end for**
19: **return** $\{\phi_j\}, \{\theta_i\}$ (updated models)

approach employs a bi-level updating strategy, where client and cluster models are updated differently.

The client model parameters $\{\theta_i\}, i \in \mathcal{C}$ are divided into two groups: cluster-shared parameters $\{\phi_i\}$ and locally-private parameters $\{h_i\}$. The former are updated and transferred to the cluster devices, while the latter are retained on the client devices for personalization. For instance, in a recurrent neural network used for sequential prediction, the recurrent blocks can be designated as $\{\phi_i\}$ to capture the shared characteristics among cluster members. Meanwhile, the local client parameters $\{h_i\}$ can be a dense layer that maps the output of the recurrent layer to the prediction.

During each communication round, client models obtain the most recent client model $\theta_i^{(t)}$ by minimizing the local objective defined in Equation 2 (line 8, Alg. 2), where τ denotes the number of local updates for client models. Specifically, the local objective F_i is defined as:

$$\min_{\theta_i} F_i(\theta_i) \text{ with } F_i(\theta_i) \coloneqq \mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}}) + \lambda \mathcal{L}_p(\varphi_i, \hat{\mathcal{Y}}) \quad (2)$$

Here, $\mathcal{L}$ is a local loss function, such as Mean Squared Error, and $\mathcal{L}_p$ is an additional loss function with respect to the client STL property φ_i. The hyperparameter λ is used to control the strength of the property loss.

We employ stochastic gradient descent (SGD) as the optimization algorithm to update the neural network models (line 8, Alg. 2), as shown in Equation 3, where η denotes the step size for the gradient descent. The SGD update rule can be substituted with any other gradient descent-based algorithm. After local client updating, the shared layers $\{\phi_i\}$ are uploaded to the cluster model (line 10, Alg 2).

$$(\theta_i^{(t)}) \leftarrow \text{SGD}(\theta_i^{(t)}, F_i, \eta) \quad (3)$$

During the cluster updating rounds, each cluster model aggregates the updated layers from its members using the function $G(\cdot)$ (line 13, Alg. 2). In FedSTL, clusters compute $G(\cdot)$ directly as a weighted average. Finally, in order to further exploit the shared logic reasoning property among cluster devices, the framework performs κ rounds of updates on the cluster models while enforcing the inducted STL constraint $\{\varphi_j\}$, where φ_j is a temporal reasoning property inducted for the j^{th} cluster. This is done by optimizing the objective specified in Equation 4, and the process is described in line 15 of Alg. 2.

$$\begin{gathered}\min_{\phi_j} F_j(\phi_j), \text{ with } F_j(\phi_j) \coloneqq \mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}}) + \lambda \mathcal{L}_p(\varphi_j, \hat{\mathcal{Y}}) \\ (\phi_j^{(t)}) \leftarrow \text{SGD}(\phi_j^{(t)}, F_j, \eta_j)\end{gathered} \quad (4)$$

Importantly, synchronizing client parameters with clusters does not require client personalization on $\{h_i\}$ to be completed first, as the personalized layers are not shared among clients. This means that client models can continue to perform local personalization, even if they are selected to participate in a given communication round.

Evaluation

Our evaluations revolve around the following primary objectives: (1) Enhancing personalized FL by incorporating *locally-specific logic reasoning* properties into real-world

Method	RNN		GRU		LSTM		Transformer	
	MSE	γ	MSE	γ	MSE	γ	MSE	γ
FedAvg	.128±.032	78.96±1.03	.154±.031	80.51±0.75	.126±.034	81.80±0.91	.588±.005	78.01±0.62
FedProx	.128±.032	78.86±1.02	.154±.032	80.43±0.74	.126±.034	81.93±0.89	.588±.005	78.05±0.63
FedRep	.164±.033	80.04±1.00	.279±.029	80.08±0.01	.214±.031	81.08±0.08	.929±.004	57.01±0.48
Ditto	.124±.031	79.17±0.01	.153±.032	80.41±0.74	.128±.035	81.48±0.84	.591±.005	78.05±0.63
IFCA	.117±.031	77.89±0.95	.140±.034	77.47±0.75	.121±.034	80.41±0.87	.063±.004	73.43±0.56
IFCA-S	.107±.032	78.89±0.96	.134±.033	77.66±0.80	.110±.035	81.51±0.90	.061±.004	72.79±0.56
FedSTL-S	.096±.026	81.67±0.93	.148±.031	81.71±0.76	.111±.030	83.44±0.87	**.025±.003**	77.03±0.93
FedSTL	.095±.026	81.70±0.98	.152±.031	81.83±0.71	.119±.031	83.32±0.92	.029±.003	78.99±0.66
FedSTL-T	**.076±.022**	**100.0±0.00**	**.118±.027**	**100.0±0.00**	**.099±.026**	**100.0±0.00**	.287±.013	**100.0±0.00**

Table 2: Comparison on MSE and locally-distinctive property satisfaction.

sequential prediction datasets. (2) Integrating *intra-task symbolic reasoning* into multitasking personalized FL training objectives and assessing its effectiveness in diverse client configurations. (3) Highlighting the advantages of client model personalization enabled by our method, and comparing the results with other existing FL approaches. The experiments were conducted on a machine equipped with an Intel Core i9-10850K CPU and an NVIDIA GeForce RTX 3070 GPU. The operating system used was Ubuntu 18.04.

Experiment Setup We evaluate the performance of FedSTL in two distinct scenarios: (1) a synthetic multivariate large-scale smart city dataset, and (2) a real-world univariate highway traffic volume dataset. For baseline comparisons, we utilize three different backbone networks: a vanilla RNN, a GRU model, a transformer model, and an LSTM model. During each round of FL communication, we randomly select 10% of the client devices to participate. For all the conducted experiments and algorithms, we use SGD with consistent learning rates and a batch size of 64.

Baseline Methods We compare the performance of FedSTL with the following methods: (1) *FedAvg* (McMahan et al. 2017) is a widely-used FL algorithm that trains a global model by aggregating the weighted average of client models; (2) *FedProx* (Li et al. 2020) is a generalization of FedAvg that addresses system and statistical heterogeneity with a reparametrization technique; (3) *FedRep* (Collins et al. 2021) is a personalized FL algorithm that learns shared global representations with unique local heads for each client; (4) *Ditto* (Li et al. 2021a) is a personalized FL method that incorporates regularization techniques to enhance the fairness and robustness; (5) *IFCA* (Ghosh et al. 2020) is a clustering FL algorithm that iteratively groups participating clients based on their training goals to promote collaboration among clients with similar objectives. We set the number of local epochs to 10 for FedAvg, FedProx, FedRep (with 8 head epochs), Ditto, and IFCA. Additionally, for FedSTL, we employ 6 local epochs and 4 cluster training epochs.

Evaluation Metrics We utilize mean squared error (MSE) as our metric to evaluate the network performance. In addition, we introduce a measure called the satisfaction rate (γ) to evaluate the impact of FedSTL on client property satisfaction. Specifically, we define γ as the percentage of network predictions, denoted by $\hat{\mathcal{Y}} = (y_{n+1}, \ldots, y_{n+m})$, that satisfy a given property φ, induced by the input sequence $\mathcal{X} = (x_1, \ldots, x_n)$. This allows us to quantify the degree to which the predicted sequence $\hat{\mathcal{Y}}$ adheres to the specified property φ based on the input sequence $\mathcal{X}$.

Enhancing Locally-Specific Logic Reasoning Properties In our first task, we enhance personalized federated learning by *incorporating locally distinct temporal properties* using real highway traffic data. We specifically focus on the operational range property, as outlined in Table 1, which captures important aspects of the traffic volume dynamics for each client during two-hour windows. We obtain a publicly available dataset from the Federal Highway Administration (FHWA 2016) and preprocess hourly traffic volume from 15 states. Further, we design a testing scenario where a neural network is trained to predict the traffic volume for the next 24 consecutive hours based on the past traffic volume at a location over the previous five days.

Enhancing Intra-Task Symbolic Properties In our second task, the objective is to enhance personalized federated learning (FL) by incorporating *intra-task symbolic reasoning properties*, where the two variables were the number of vehicles on the road and the occupancy of the same road. To achieve this, we create a simulated dataset using SUMO (Simulation of Urban MObility) (Krajzewicz et al. 2002), a large-scale open-source road traffic simulator. The learning objective in this task is to predict a multivariate traffic and pollution scenario. Moreover, we focus on diverse road types and consider a traffic scenario that includes cars, trucks, and motorcycles. From the available road segments, we select 100 segments to serve as FL clients. For each client, we record various features, including vehicle counts, road occupancy, mean speed, carbon dioxide emission, average fuel consumption, and noise emission.

Results and Discussion Table 2 presents the results of FedSTL in enhancing locally distinctive reasoning proper-

ties, where "-S" indicates the evaluation of the cluster model, and "-T" indicates the evaluation of the teacher model. The performance of FedSTL surpasses that of various other FL methods, both personalized and non-personalized. Specifically for FedSTL, the "Cluster" row demonstrates the effectiveness of our clustering method, while the "Client" row represents the performance on our client devices prior to prediction correction by the teacher. In contrast, the "Teacher" row shows the framework's performance after the prediction is corrected by the teacher. Figure 3 illustrates the comparison results on MSE by enhancing intra-task reasoning properties. The flowpipe representation is used to indicate the error bars.

We observed a significant improvement in the MSE for RNN models, with up to a 54% reduction compared to the baseline. Similarly, GRU models exhibited a 53.8% lower MSE, and LSTM models achieved up to a 53.6% lower MSE. Furthermore, the teacher model within the FedSTL framework consistently corrected predictions with a 100% satisfaction rate across all cases.

By enhancing locally-distinct properties for FL clients, we observe a substantial improvement in the model's prediction performance, as indicated by both the MSE and satisfaction rate metrics. When comparing FedAvg, FedProx, and Ditto, we find that their MSE values are generally similar. However, FedRep exhibits relatively poorer performance, while IFCA consistently outperforms the other methods. When locally-distinct properties are incorporated, FedSTL surpasses IFCA in terms of predictive accuracy. Notably, the teacher component of FedSTL demonstrates the best performance, with significantly lower MSE and a higher satisfaction rate for the properties. These findings indicate a promising trend: by correcting predictions based on localized properties, we can achieve a significant improvement in the model's accuracy. Additionally, in the context of enhancing intra-task properties, both FedSTL and IFCA show superior performance compared to other baselines. These results underscore the importance of aligning client training objectives to enhance the overall performance of the model.

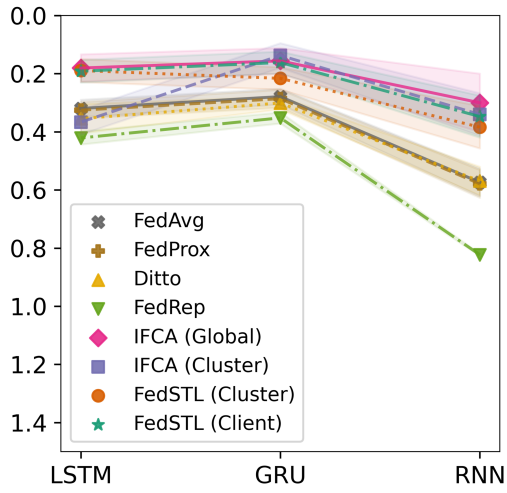

Figure 3: Comparison on MSE with enhancing intra-task reasoning properties. A higher position on the y-axis indicates a smaller MSE value.

Related Work

In contrast to traditional FL frameworks, personalized FL prioritizes the training of local models tailored to individual clients, rather than relying on a single global model that performs similarly across all clients (Tan et al. 2022; Fallah, Mokhtari, and Ozdaglar 2020; Collins et al. 2021; Ghosh et al. 2020; Arivazhagan et al. 2019; Mansour et al. 2020). Deng et al. (Deng, Kamani, and Mahdavi 2020) highlight the significance of personalization in FL algorithms, particularly when dealing with non-i.i.d. client datasets. In light of this, our work focuses on investigating the potential benefits of incorporating symbolic reasoning through formal specification to enhance personalized FL algorithms. Specifically, we leverage rigorous and formal logic properties to improve the predictions of neural networks. This approach aligns with the concept of informed machine learning, which integrates auxiliary domain knowledge into the machine learning framework, as emphasized in a comprehensive survey by Von Rueden et al. (Von Rueden et al. 2021). Generally, such methods are critical in improving the performance of data-driven models across various aspects, as shown in previous related works (Muralidhar et al. 2018; Ma et al. 2021b; Diligenti, Roychowdhury, and Gori 2017; Jia et al. 2021; Ma et al. 2020; Hu et al. 2020; An and Ma 2023).

Summary and Future Work

Personalized FL methods have been developed to address the challenge of heterogeneous client devices. However, these methods have largely overlooked the potential of symbolic reasoning in tackling this issue. To bridge this gap, our study explores the effectiveness of incorporating symbolic reasoning into personalized FL. Our evaluation results significantly improve client prediction error and property satisfaction when leveraging induced client device properties. Furthermore, our observation indicates that enhancing the satisfaction of properties also reduces error rates. This result shows the promising potential of equipping deep learning models with symbolic reasoning capabilities.

Moreover, while our main concentration was on evaluating prediction accuracy and property satisfaction in AIoT, there is scope to extend this work to other domains. First, AI in healthcare presents unique challenges in privacy and safety. Future research could investigate the integration of symbolic reasoning into AI algorithms to foster safe and robust models in healthcare settings. Second, in smart energy systems, symbolic reasoning can be holistically integrated to augment real-time data analytics with rule-based decision making. Lastly, by regulating data-driven models with logic properties, post-hoc interpretability naturally emerges. Future work could focus on making models more transparent and trustworthy by harnessing the advantages of symbolic reasoning.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Award Numbers 2028001 and 2220401, AFOSR under FA9550-23-1-0135, and DARPA under FA8750-23-C-0518.

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)*, ICCPS '23, 274–275. New York, NY, USA: Association for Computing Machinery. ISBN 9798400700361.

Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

Bartocci, E.; Mateis, C.; Nesterini, E.; and Nickovic, D. 2022. Survey on mining signal temporal logic specifications. *Information and Computation*, 104957.

Chen, D.; Gao, D.; Kuang, W.; Li, Y.; and Ding, B. 2022. pFL-Bench: A Comprehensive Benchmark for Personalized Federated Learning. *arXiv preprint arXiv:2206.03655*.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, 2089–2099. PMLR.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*.

Diligenti, M.; Roychowdhury, S.; and Gori, M. 2017. Integrating prior knowledge into deep learning. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, 920–923. IEEE.

Donzé, A.; and Maler, O. 2010. Robust satisfaction of temporal logic over real-valued signals. In *Formal Modeling and Analysis of Timed Systems: 8th International Conference, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings 8*, 92–106. Springer.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

FHWA. 2016. Highway Performance Monitoring System Field Manual. https://www.fhwa.dot.gov/policyinformation/hpms/fieldmanual/hpms_field_manual_dec2016.pdf. Accessed: 2024-01-24.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2020. Harnessing Deep Neural Networks with Logic Rules. arXiv:1603.06318.

Jha, S.; Tiwari, A.; Seshia, S. A.; Sahai, T.; and Shankar, N. 2017. Telex: Passive stl learning using only positive examples. In *International Conference on Runtime Verification*, 208–224. Springer.

Jia, X.; Willard, J.; Karpatne, A.; Read, J. S.; Zwart, J. A.; Steinbach, M.; and Kumar, V. 2021. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science*, 2(3): 1–26.

Krajzewicz, D.; Hertkorn, G.; Rössel, C.; and Wagner, P. 2002. SUMO (Simulation of Urban MObility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, 183–187.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021a. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, 6357–6368. PMLR.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; and Dou, Q. 2021b. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*.

Ma, M.; Bartocci, E.; Lifland, E.; Stankovic, J. A.; and Feng, L. 2021a. A novel spatial–temporal specification-based monitoring system for smart cities. *IEEE Internet of Things Journal*, 8(15): 11793–11806.

Ma, M.; Gao, J.; Feng, L.; and Stankovic, J. 2020. STLnet: Signal temporal logic enforced multivariate recurrent neural networks. *Advances in Neural Information Processing Systems*, 33: 14604–14614.

Ma, M.; Preum, S. M.; Ahmed, M. Y.; Tärneberg, W.; Hendawi, A.; and Stankovic, J. A. 2019. Data sets, modeling, and decision making in smart cities: A survey. *ACM Transactions on Cyber-Physical Systems*, 4(2): 1–28.

Ma, M.; Preum, S. M.; and Stankovic, J. A. 2017. Cityguard: A watchdog for safety-aware conflict detection in smart cities. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 259–270.

Ma, M.; Stankovic, J.; Bartocci, E.; and Feng, L. 2021b. Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(5s): 1–25.

Ma, M.; Stankovic, J. A.; and Feng, L. 2018. Cityresolver: a decision support system for conflict resolution in smart cities. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 55–64. IEEE.

Ma, M.; Stankovic, J. A.; and Feng, L. 2021. Toward formal methods for smart cities. *Computer*, 54(9): 39–48.

Maler, O.; and Nickovic, D. 2004. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, 152–166. Springer.

Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*.

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.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic federated learning. In *International Conference on Machine Learning*, 4615–4625. PMLR.

Muralidhar, N.; Islam, M. R.; Marwah, M.; Karpatne, A.; and Ramakrishnan, N. 2018. Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE international conference on big data (big data)*, 36–45. IEEE.

Nguyen, D. C.; Ding, M.; Pathirana, P. N.; Seneviratne, A.; Li, J.; and Poor, H. V. 2021. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3): 1622–1658.

Preum, S. M.; Munir, S.; Ma, M.; Yasar, M. S.; Stone, D. J.; Williams, R.; Alemzadeh, H.; and Stankovic, J. A. 2021. A review of cognitive assistants for healthcare: Trends, prospects, and future directions. *ACM Computing Surveys (CSUR)*, 53(6): 1–37.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Von Rueden, L.; Mayer, S.; Beckh, K.; Georgiev, B.; Giesselbach, S.; Heese, R.; Kirsch, B.; Pfrommer, J.; Pick, A.; Ramamurthy, R.; et al. 2021. Informed Machine Learning–A taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 614–633.

Yu, T.; Bagdasaryan, E.; and Shmatikov, V. 2020. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*.